



UNIVERSIDAD POLITÉCNICA DE MADRID

FACULTAD DE INFORMÁTICA

TRABAJO FIN DE CARRERA

**APLICACIÓN DEL TRACKING BASADO EN PUNTOS
CARACTERÍSTICOS A LA CINEMOMETRÍA DEL
TRÁFICO**

AUTOR: Noé Casas Manzanares

TUTOR: Luis Baumela Molina

Noviembre 2007

Índice general

1. Introducción y objetivos	1
1.1. Aplicación de técnicas de seguimiento visual	1
1.2. Estructura de la memoria	3
2. Aproximaciones habituales	5
2.1. Diferencia de imágenes	5
2.2. Flujo óptico	6
2.3. <i>Active Contours</i>	7
2.4. Seguimiento basado en puntos característicos	9
2.5. Seguimiento basado en modelos	9
3. Solución implementada	11
3.1. Esquema general	11
3.2. Modelo de características	13
3.2.1. Extracción de características	14
3.2.2. Modelado de cámara	16
3.2.3. Filtro de Kalman	23
3.3. Modelo de conectividad	29
3.3.1. Modelo del fondo	34
3.4. Extracción de vehículos	35
4. Resultados	37
4.1. Experimentos con secuencias sintéticas	37
4.1.1. Objetivos	38
4.1.2. Composición de las escenas	38
4.1.3. Calibración del plano del suelo	39
4.1.4. Experimentos	41
4.2. Experimentos con secuencias reales	47
4.2.1. Objetivos	47
4.2.2. Composición de las escenas	47
4.2.3. Calibración del plano del suelo	47
4.2.4. Experimentos	48
4.3. Análisis de los resultados	50
4.3.1. Captura	50

4.3.2.	Agrupamiento de características	50
4.3.3.	Estimación de la velocidad de los vehículos	51
5.	Conclusiones y líneas futuras	53
5.1.	Conclusiones	53
5.2.	Líneas futuras	54
5.2.1.	Implementación <i>realtime</i>	54
5.2.2.	Mejora de la segmentación	55
5.2.3.	Autocalibración	55
5.2.4.	<i>Structure from motion</i>	55
5.2.5.	<i>Object recognition</i>	56
5.2.6.	Infraestructura de monitorización de tráfico	56
	Bibliografía	59

Índice de figuras

2.1. <i>Active contours</i> aplicado al seguimiento de tráfico.	8
2.2. Modelos tridimensionales de vehículos genéricos.	10
2.3. Encaje del modelo tridimensional a un vehículo de la escena.	10
3.1. Esquema general.	11
3.2. Esquema general detallado.	12
3.3. Actualización del modelo de características.	13
3.4. Esquema del proceso de adquisición de imágenes.	16
3.5. Función del modelo de proyección.	17
3.6. Modelo de proyección.	17
3.7. Esquema del modelo de proyección.	18
3.8. Paso del CCD a una matriz de píxels en memoria	19
3.9. Cambio de sistema de referencia en el modelo extrínseco	20
3.10. Esquema general del bucle iterativo del filtro de Kalman.	23
3.11. Bucle iterativo del filtro de Kalman.	27
3.12. Desviación de las proyecciones de las esquinas.	29
3.13. Actualización del modelo de conectividad.	30
3.14. Fotograma actual.	31
3.15. Modelo del fondo.	31
3.16. Diferencia del fotograma actual con el fondo.	32
3.17. Umbralización de la diferencia con el fondo.	32
3.18. Componentes conexas en la umbralización de la resta con el fondo. . .	33
3.19. Agrupamiento de esquinas dentro de componentes conexas.	33
3.20. Grafo de conectividad entre las esquinas.	34
3.21. Modelo del fondo en dos instantes distintos de la secuencia real. . . .	35
4.1. Escena sintética.	38
4.2. Escena sintética desde las localizaciones de la lateral y cenital.	39
4.3. Plantillas de calibración de la escena sintética.	40
4.4. Snapshot del tracking de la secuencia del test 1.	41
4.5. Velocidades medidas en el test 1, vista lateral.	42
4.6. Velocidades medidas en el test 1, vista cenital.	42
4.7. Snapshot del tracking de la secuencia del test 2.	43
4.8. Velocidades medidas en el test 2, vista lateral.	43
4.9. Velocidades medidas en el test 2, vista cenital.	44

4.10. Snapshot del tracking de la secuencia del test 3.	45
4.11. Histograma de las estimaciones de velocidad de los vehículos.	45
4.12. Histograma de las estimaciones de velocidad de los vehículos.	46
4.13. Calibración “a mano” de las escenas reales.	48
4.14. Fotogramas de la secuencia real.	48
4.15. Error en la segmentación de vehículos en la secuencia real.	49
4.16. Modelo del fondo en dos instantes distintos de la secuencia real.	49
4.17. Proyección de la silueta del vehículo en el plano del suelo.	51
5.1. Información de tráfico solapada a <i>Google maps</i> en la web de la DGT.	57

Capítulo 1

Introducción y objetivos

El hecho de que el exceso de velocidad sea una de las principales causas de accidentes en carretera hace que la medición y control de la velocidad del tráfico rodado cobre cada día mayor importancia.

Para ejercer dicho control, la *Dirección General de Tráfico* instala en las carreteras de todo el territorio nacional aparatos electrónicos, llamados ***cinemómetros***, popularmente conocidos como “radares”, capaces de medir la velocidad de los vehículos.

Los radares usados habitualmente para la medición de la velocidad del tráfico suelen ser emisores/receptores de trenes de pulsos de ondas de radio que utilizan el denominado *efecto Doppler*, por el que los objetos que pasen delante del haz de ondas emitido, alteraran la frecuencia de su eco, positivamente si se acercan o negativamente si se alejan, de una manera predecible. Por tanto, conocida la frecuencia de las ondas emitidas, puede saberse la velocidad de los objetos que se interponen en su camino.

El objetivo del presente trabajo es el desarrollo de un cinemómetro de tráfico que haga uso únicamente de técnicas de visión artificial, es decir, implementar un software que tome como entrada una secuencia de vídeo tomada de una cámara de tráfico y sea capaz de generar como salida la misma información que un cinemómetro basado en el efecto Doppler, es decir, la velocidad del vehículo que en ese momento circula dentro de su campo de acción.

1.1. Aplicación de técnicas de seguimiento visual

A diferencia del radar basado en el efecto Doppler, que es capaz de medir la velocidad de los vehículos de manera *instantánea*, nosotros necesitaremos *percibir* los vehículos durante un periodo de tiempo y observar el cambio en su posición para obtener la misma información, ya que un solo fotograma no nos proporciona información

suficiente sobre el movimiento del coche.

Así, deberemos *seguir* a los vehículos durante un determinado número de frames, tras los que podremos estimar la velocidad a la que ha circulado a lo largo de ese periodo. En el *argot* de la visión computacional, se conoce a dicho problema como el **problema de seguimiento**.

El problema del seguimiento o *tracking* es uno de los más antiguos en el área de la visión artificial y consiste en conocer en cada instante el “estado” (posición, velocidad, aceleración, etc) de elementos que aparezcan en una secuencia de imágenes.

Existen ciertas condiciones (restricciones) que hacen más apropiado el uso de ciertas técnicas; dichas condiciones son:

- Movilidad de la cámara: si se trata de una cámara fija o por el contrario puede encontrarse en movimiento.
- Número de objetos a seguir: si se busca seguir a un solo objeto o a varios.

En el caso de seguimiento de vehículos mediante cámaras de tráfico, el problema se trata de seguir a **múltiples objetos** teniendo una **cámara fija**.

Distintas aproximaciones al problema de seguimiento se centran en diferentes *propiedades* de los objetos, que les permiten diferenciarse del resto de la escena; así, por ejemplo, en técnicas de seguimiento basadas en flujo óptico se utiliza el propio movimiento de los objetos relativo al resto de la imagen, mientras que técnicas como *active contours* se centran en el *contorno* de los objetos a seguir, y en técnicas basadas en características, el centro de atención son características locales de los objetos a seguir, como texturas o esquinas. El proceso de seleccionar los aspectos de los objetos que interesen al tracker y utilizarlos como representación suya en nuestra “realidad simplificada” permite **modelizar** los objetos a seguir en base a una simplificación de su *estado* observable.

La aproximación habitual para implementar un *tracker*, una vez formulada la modelización del estado de los objetos a usar, es proceder a la detección de los objetos a seguir, haciendo uso y posteriormente actualizando la información acerca de la escena en frames previos. De esta forma, el algoritmo seguidor va manteniendo a lo largo de los fotogramas de la secuencia una “imagen” de lo que cree que contiene la escena.

Dicha información sobre lo que ocurre en la secuencia es procesada para extraer datos de interés, por ejemplo, en nuestro caso extraeremos cifras sobre la velocidad de los vehículos que circulen por la carretera.

1.2. Estructura de la memoria

Tras esta breve introducción al problema de medición de la velocidad del tráfico rodado mediante técnicas de visión artificial, en el capítulo 2 exploraremos las aproximaciones más comunes al *tracking* de objetos, tales como el flujo óptico o el seguimiento basado en modelos.

A continuación, en el capítulo 3 procederemos a exponer la estrategia desarrollada en el presente proyecto, aumentando progresivamente el nivel de detalle.

Más adelante, en el capítulo 4 presentaremos los resultados obtenidos con el *tracker* implementado, tanto con imágenes sintéticas preparadas al efecto como con imágenes reales tomadas de cámaras de tráfico.

Por último, en el capítulo 5 extraeremos las conclusiones apropiadas y atisbaremos las posibles líneas a seguir para extender la funcionalidad desarrollada.

Capítulo 2

Aproximaciones habituales

El problema de seguimiento es uno de los más tratados dentro del campo de la visión artificial, por lo que a lo largo de los años han sido muchas las aproximaciones desarrolladas. En este capítulo abordaremos de manera general las soluciones más utilizadas para tratar concretamente con el problema del seguimiento de vehículos.

2.1. Diferencia de imágenes

La diferencia de imágenes es uno de los métodos más antiguos aplicados al problema de seguimiento. Para poder aplicarlo, es necesario disponer de una imagen estática del fondo que no contenga ningún elemento ajeno a él. De esta manera, a cada fotograma de la secuencia se le resta la imagen del fondo, umbralizándose la imagen resultante, de tal modo que se obtiene una *máscara booleana* que nos permite diferenciar los elementos de primer plano.

Este método es muy sensible, adoleciendo de bastantes problemas:

- Cambios en la iluminación del fondo: en escenarios exteriores, son frecuentes los cambios de iluminación según va avanzando el día y cambiando las condiciones climatológicas. De esta manera, la imagen inicial del fondo, deja de tener validez. Para solventar este problema, se suele mantener un modelo del fondo que vaya evolucionando lentamente pero que no se vea influido por los elementos de primer plano que aparezcan en escena.
- Elementos dinámicos del fondo: fondos que contienen elementos cuyo aspecto pueda variar (por ejemplo, las hojas de los árboles con el viento), son especialmente inadecuados para este tipo de algoritmos.
- Bajo contraste de los elementos de primer plano: el paso de la umbralización de la resta del fotograma con la imagen del fondo, nos obliga a establecer un

umbral fijo, de manera que los píxels cuyo valor de luminosidad no supere dicho valor, serán tomados como fondo. Los objetos que presentan un contraste bajo respecto al fondo son normalmente asimilados a éste al umbralizar debido a este hecho.

- Sombras: este tipo de algoritmos son mecánicos, diferenciando el fondo de lo que no lo es, y por tanto, asimilando las sombras de los objetos como elementos de primer plano.

Este algoritmo se trata más bien de un mecanismo de *segmentación* en vez de de seguimiento, por lo que suele ser usado como apoyo a algoritmos de *tracking* más sofisticados.

2.2. Flujo óptico

Se da el nombre de campo de movimiento o ***motion field*** al campo vectorial que expresa el movimiento de cada punto del espacio. Llamamos **flujo óptico** a la proyección del *motion field* sobre el plano imagen. Desde un punto de más aplicable, el flujo óptico puede aproximarse como la distribución de velocidades aparentes de los patrones de luminosidad a lo largo de una secuencia de imágenes ([8]).

Para poder estimar el flujo óptico, es necesario introducir algún tipo de restricción; normalmente, las restricciones que suelen aplicarse son

- **Ecuación de constancia de la luminosidad** (*brightness constancy equation*): la luminosidad de un píxel determinado que corresponde a una superficie no cambia cuando se desplaza de una imagen a la siguiente, es decir, para cada punto del plano imagen debe cumplirse que:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (2.1)$$

o también expresado como

$$\frac{dI}{dt} = 0 \quad (2.2)$$

siendo $I(x, y, t)$ la luminosidad del punto (x, y) de la imagen en el instante t . Desarrollando dicha ecuación, se obtiene la forma habitual de la restricción:

$$I_x u + I_y v = -I_t \quad (2.3)$$

donde I_x , I_y y I_t son los gradientes de la imagen y (u, v) las componentes del flujo óptico.

- Restricción de suavidad (*smoothness constraint*) ([13, 14]): el flujo óptico (u, v) debe variar de manera suave a lo largo del plano imagen (situación que normalmente no se da ante objetos ocluidos por otros).

Dichas restricciones permiten, mediante técnicas diferenciales o basadas en características ([1]), el cálculo del flujo óptico, sin embargo lo hacen especialmente inapropiado para ciertas situaciones:

- Cambios rápidos de iluminación, siendo interpretados como movimiento.
- Objetos homogéneos, sin textura, ya que generan flujo óptico nulo.
- Objetos no rígidos, interpretándose las “deformaciones” del objeto como movimiento.

El cálculo del flujo óptico ha sido aplicado con éxito a varios dominios:

- *Tracking* ([15]): seguimiento de objetos en movimiento.
- Segmentación por movimiento: uso de la información del flujo óptico para segmentar objetos en movimiento.
- *Ego-motion* ([21, 22]): determinación del movimiento de un objeto a partir de una secuencia de imágenes tomadas desde una cámara montada en el propio objeto.
- *Structure from motion*: determinación de la estructura tridimensional de los objetos a partir del flujo óptico durante una secuencia de imágenes.

2.3. *Active Contours*

En el algoritmo *active contours*, también llamado “*snakes*” ([10, 4]), el objeto a seguir viene representado únicamente por su borde o contorno, que es modelizado mediante una *spline* (véase la figura 2.1, donde puede verse la técnica *active contours* aplicada al dominio del seguimiento de tráfico en carretera). De esta manera es en cierto modo una “generalización conceptual” de los mecanismos de detección de líneas y esquinas, tan usados a lo largo de la historia de la visión computacional.

Las splines que representan los bordes de las figuras no surgen de la nada, sino de la minimización de una función de energía definida por nosotros. La definición de esta función de energía, para la que es necesaria la imposición de restricciones basadas en el conocimiento previo *a priori* que se posee acerca de los objetos que se pretende seguir, es la que permite que tenga como mínimo local los bordes de las figuras:

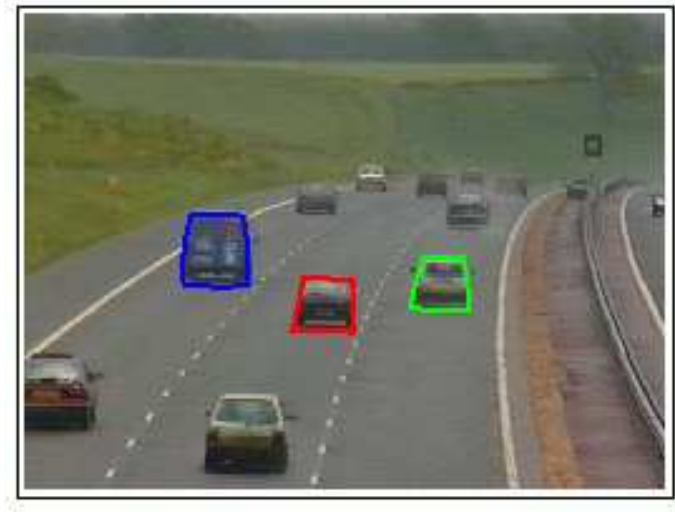


Figura 2.1: *Active contours* aplicado al seguimiento de tráfico.

$$\begin{aligned}
 E_{snake}^* &= \int_0^1 E_{snake}(v(s)) ds = \\
 &= \int_0^1 [E_{int}(v(s)) + E_{image}(v(s))] + E_{con}(v(s)) ds
 \end{aligned} \tag{2.4}$$

donde:

- E_{int} representa la energía interna de la spline debido a la curvatura y se define como $(\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)/2$, de manera que ajustando los términos $\alpha(s)$ y $\beta(s)$ se altera el comportamiento de la misma.
- E_{image} expresa las “fuerzas de la imagen”, es decir, es la expresión que hace que la spline se ajuste a los bordes de las figuras que nos interesan. Normalmente esta expresión se escoge como el gradiente de la imagen, de manera que la curva se ajuste a los bordes marcados por el mismo, aunque también existen variaciones que pretenden asegurar la conectividad de las regiones de mínima energía.
- E_{con} define las restricciones impuestas de manera externa, lo que nos permite establece la “forma” que tendrá el spline, aprovechando el conocimiento *a priori* de que dispongamos acerca del contorno esperado de los objetos a seguir.

La técnica de *active contours* se ha aplicado con éxito a muchos dominios, incluyendo el del seguimiento del tráfico rodado (véase figura 2.1), ya que es tolerante a oclusiones y ofrece un mecanismo visualmente intuitivo para el observador humano,

sin embargo adolece de ciertos problemas como la necesidad del conocimiento a priori de la forma de los objetos a seguir, así como la invariabilidad de la misma a lo largo de la secuencia, lo que en imágenes reales pocas veces se da.

El éxito de una aproximación basada en *active contours* se basa en la elección apropiada de las funciones que determinan la energía de la curva, así como de los parámetros de las mismas.

2.4. Seguimiento basado en puntos característicos

También llamado seguimiento basado en características o en *features* (*feature-based tracking*), se basa en el seguimiento individualizado de características locales de los objetos de interés, tales como puntos, líneas o texturas; ello lo diferencia claramente de algunas de las aproximaciones anteriormente citadas, que pretenden seguir el objeto completo.

Lo que se consigue mediante este tipo de técnicas es eliminar la complejidad del proceso de seguimiento (ya que se siguen características elementales, fácilmente identificables y distinguibles), pero haciendo necesaria una “capa de mayor abstracción” que sea capaz de **agrupar** las características pertenecientes a un mismo objeto.

El desarrollo del presente proyecto se basa en este tipo de técnicas, que serán exploradas en mayor profusión a lo largo de esta memoria.

2.5. Seguimiento basado en modelos

Las técnicas de seguimiento basadas en modelos ([11]) tratan de ajustar los vehículos que circular en la escena a modelos tridimensionales de vehículos genéricos (figura 2.2).

Para ello, se sigue a los vehículos mediante *tracking* basado en puntos característicos y posteriormente, para cada vehículo, se extrae líneas de la imagen bidimensional y se intentan encajar en los modelos tridimensionales genéricos de los que se disponga; si se consigue un encaje aceptable, se asocia dicho modelo al vehículo (figura 2.3).

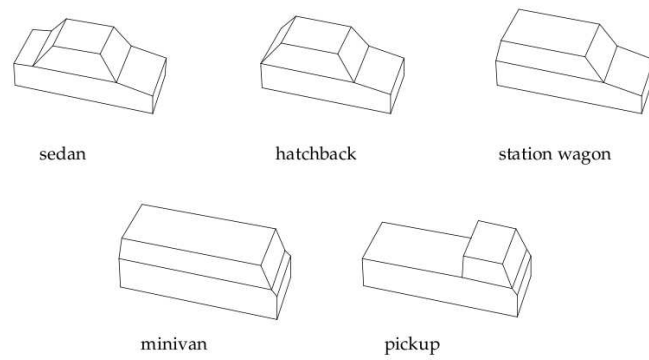


Figura 2.2: Modelos tridimensionales de vehículos genéricos.

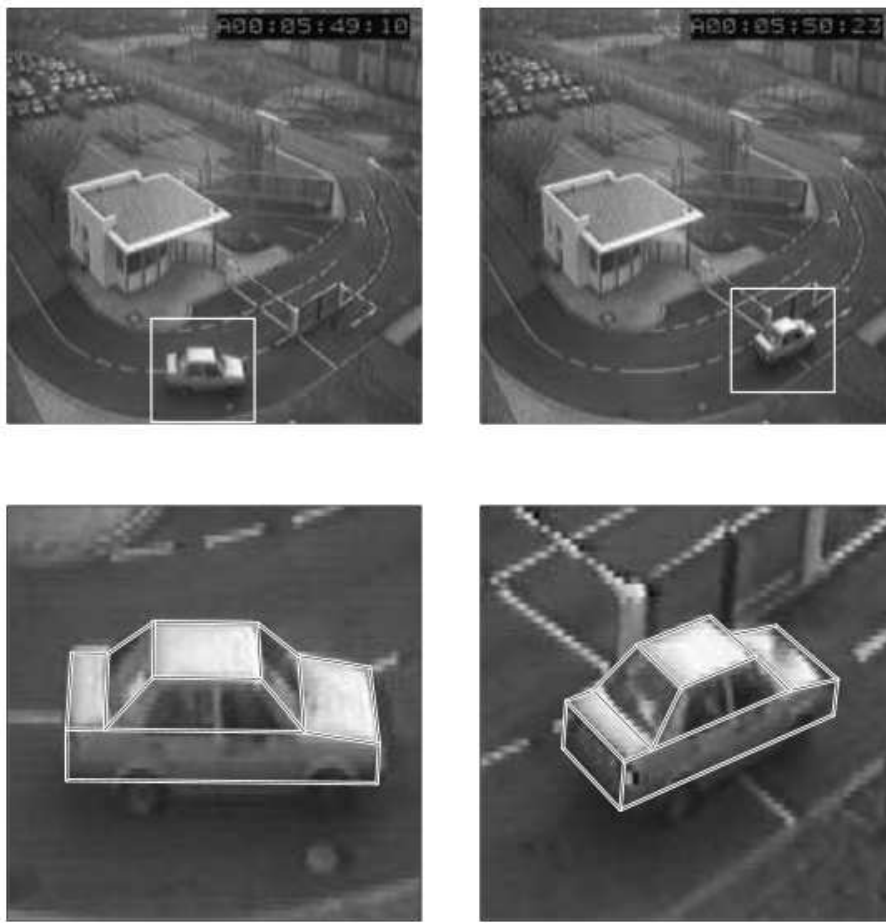


Figura 2.3: Encaje del modelo tridimensional a un vehículo de la escena.

Capítulo 3

Solución implementada

3.1. Esquema general

Como comentamos previamente en las secciones introductorias, nuestro objetivo es la construcción de un cinemómetro de tráfico, un sistema capaz de percibir el tráfico y cuantificar la velocidad a la que circulan los vehículos, utilizando únicamente técnicas de visión artificial. De esta manera, la entrada de nuestro sistema se reduciría a una secuencia de imágenes procedentes de una cámara de tráfico, y su salida debería ser un flujo de información cinemométrica de los vehículos que aparecen en la secuencia de entrada (véase figura 3.1).

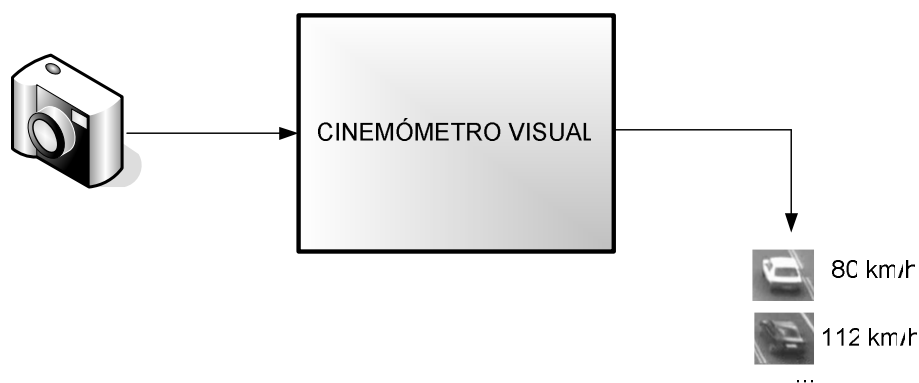


Figura 3.1: Esquema general.

Para conseguir nuestro objetivo haremos uso de técnicas de ***tracking* basado en *features* o puntos característicos**, así como de técnicas clásicas de segmentación mediante restado de imágenes.

Para poder “trasladar” la información de las imágenes a información del mundo real, supondremos que nos viene dada una transformación lineal que nos permite averiguar la proyección en el plano suelo de puntos del plano imagen, a la que

llamaremos *homografía* (véase sección 3.2.2).

Dividiremos el problema en dos subproblemas:

- El problema del seguimiento de los puntos característicos de los vehículos, al que daremos solución mediante el mantenimiento de la información sobre las *features* en lo que llamaremos el **modelo de características** (sección 3.2).
- El problema del “agrupamiento de puntos característicos pertenecientes al mismo vehículo, al que daremos solución manteniendo un grafo de conectividad entre las distintas características, al que denominaremos **modelo de conectividad** (sección 3.3).

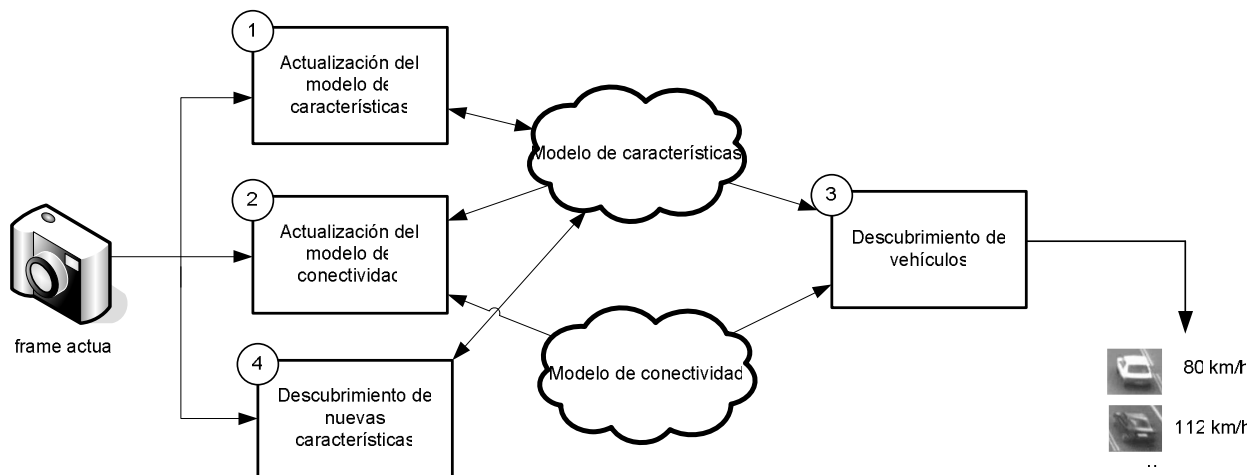


Figura 3.2: Esquema general detallado.

A partir de ambos modelos, podremos agrupar las características que pertenezcan al mismo vehículo y deducir de ellas la información cinemométrica que buscamos.

3.2. Modelo de características

El seguimiento basado en características consiste en seguir únicamente características locales de los objetos de interés, unificando posteriormente los puntos característicos pertenecientes al mismo objeto. Las características visuales que seguiremos serán **esquinas** dentro de la estructura del vehículo (véase sección 3.2.1), a las que caracterizaremos mediante una **ventana de píxels** de la imagen a su alrededor y su **posición**, **velocidad** y **aceleración** sobre el plano del suelo. A esa tupla de elementos es a lo que denominaremos “partícula”. El modelo de características se reducirá por tanto al conjunto de partículas que se *estén siguiendo* en el frame actual.

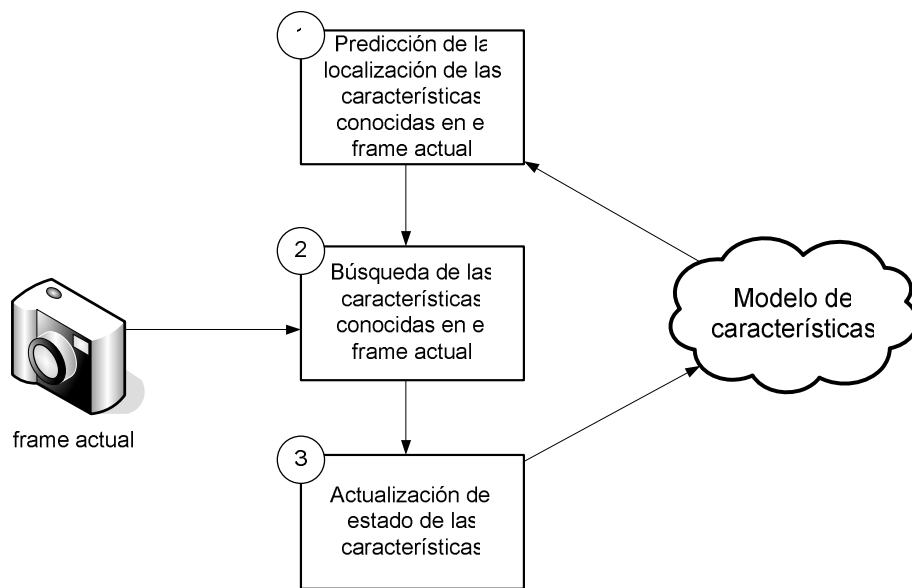


Figura 3.3: Actualización del modelo de características.

De esta manera, para mantener el modelo de características, tal como podemos observar en la figura 3.3, en cada frame deberemos actualizar el estado de las partículas que recordamos de fotogramas anteriores y buscar nuevas esquinas que añadir al modelo.

Para actualizar el estado de cada partícula del modelo...

1. Utilizando un filtro de Kalman (véase sección 3.2.3), se predice la posición del plano suelo en la que se encontrará dicha partícula en el instante actual.
2. Se traslada dicha posición al plano imagen utilizando la homografía de que disponemos (véase sección 3.2.2).
3. Se busca alrededor de dicha predicción la ventana de píxels que *se parezca más* a la ventana recordada de la partícula (aquella que maximice la correlación cruzada con ésta).

4. Si hay alguna ventana que “se parezca” lo suficiente a la que recordamos de dicha partícula, se traslada su posición al plano suelo y se actualiza el estado de la partícula

Asimismo, para localizar nuevas características que agregar al modelo...

1. Se define en la imagen una región rectangular dentro de la cual se extraen esquinas mediante el operador de Harris (sección 3.2.1).
2. Se traslada la posición de las esquinas al plano mediante la homografía de que disponemos y se incorporan dichas partículas al modelo de características.
3. Para cada una de ellas se recuerda una ventana de píxels alrededor de la partícula, que se utilizará en frames posteriores para identificarla

3.2.1. Extracción de características

Como ya se ha comentado, las características que extraeremos de los vehículos serán esquinas. Como detector de esquinas utilizaremos el propuesto por Harris y Stephens [7, 17], que utiliza la autocorrelación local del gradiente de la imagen para determinar las esquinas. Para ello, se define la matriz C para cada punto de la imagen:

$$C(x_0, y_0) = E[gg^T]_{\forall (x,y) \in \Omega} = \begin{bmatrix} \sum_{\Omega} w_i I_x^2 & \sum_{\Omega} w_i I_x I_y \\ \sum_{\Omega} w_i I_x I_y & \sum_{\Omega} w_i I_y^2 \end{bmatrix} \quad (3.1)$$

Donde I_x e I_y son los gradientes vertical y horizontal, Ω es una ventana alrededor del píxel y w_i toma los valores de una normal $N(0, \sigma^2 I)$, es decir, se pondera la cada elemento del sumatorio según un muestreo de una normal bidimensional de media 0 y covarianza $\sigma^2 I$.

Se determina si un punto es una esquina según los autovalores λ_1 y λ_2 de su matriz C asociada (al ser C una matriz semidefinida positiva sus autovalores son mayores o iguales a cero). Estas son las posibles situaciones que pueden darse:

- ventana homogénea: $\lambda_1 \approx \lambda_2$
- borde: $\lambda_1 \approx 0$, $\lambda_2 \gg 0$
- esquina: $\lambda_1 \gg 0$, $\lambda_2 \gg 0$

De esta manera, el algoritmo queda:

1. Calcular los gradientes horizontal y vertical de I: $I_x = h_{DX} * I$, $I_y = h_{DY} * I$
2. Calcular $I_x^2, I_y^2, I_x I_y$
3. $\langle I_x^2 \rangle = h_g^{n \times n} * I_x^2$ (siendo n el tamaño de Ω).
 $\langle I_y^2 \rangle = h_g^{n \times n} * I_y^2$
 $\langle I_x I_y \rangle = h_g^{n \times n} * I_x I_y$
4. $\forall \text{ pixel}(i, j), ImHarris[i, j] = \text{mínimo autovalor} \begin{bmatrix} \langle I_x^2 \rangle [i, j] & \langle I_x I_y \rangle [i, j] \\ \langle I_x I_y \rangle [i, j] & \langle I_y^2 \rangle [i, j] \end{bmatrix}$
5. $\forall \text{ pixel}(i, j)$, éste es esquina si
 - a) su autovalor mínimo es máximo local
 - b) $ImHarris[i, j] > \tau$ (umbral)

donde $h_g^{n \times n}$ es una matriz de tamaño $n \times n$ resultado de muestrear una normal de media 0 y covarianza $\sigma^2 I$

3.2.2. Modelado de cámara

Para poder establecer correspondencias entre el mundo real y los puntos de la imagen adquirida, necesitamos un modelo matemático del funcionamiento de la cámara:

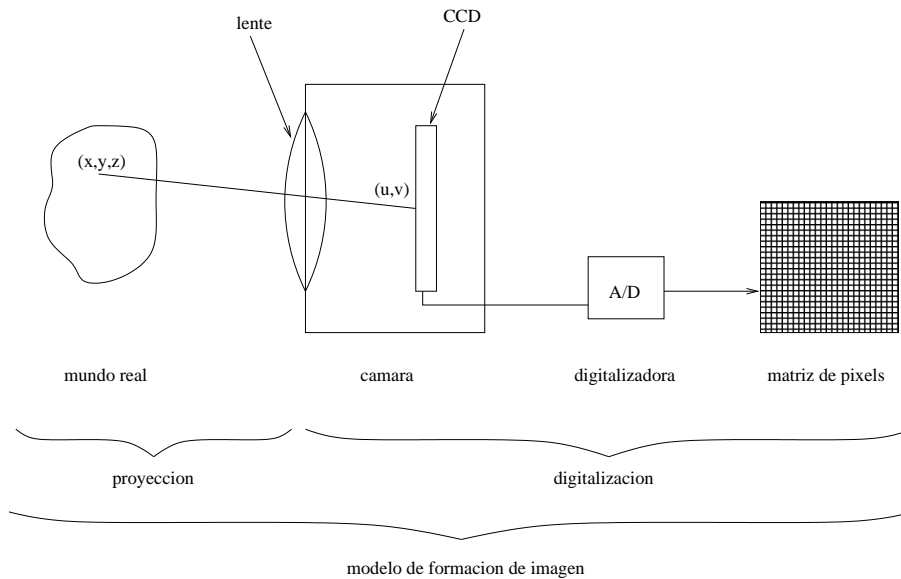


Figura 3.4: Esquema del proceso de adquisición de imágenes.

Nuestra cámara estará formada básicamente por una óptica, que será atravesada por los rayos de luz reflejados por los objetos reales y que serán refractados sobre una matriz de componentes de silicio sensibles a la luz llamada CCD (*Charge Coupled Device*), que transmitirá la información recibida a un conversor analógico digital, que nos proporcionará la matriz de píxeles de la imagen adquirida.

El modelo que utilizaremos recibe el nombre de **modelo de proyección perspectiva** o **modelo pinhole** [25, 5] que es básicamente una proyección basada en una transformación lineal proyectiva entre el espacio de puntos del mundo real y el espacio de puntos de nuestra matriz de píxeles.

El modelo *pinhole* presenta ciertas desventajas ([16]) ya que, al ser lineal, no es capaz de modelar situaciones de naturaleza no lineal como distorsiones introducidas por la lente (esférica, tangencial, radial, etc).

A continuación desarrollaremos el modelo pinhole hasta llegar a las ecuaciones que nos permitan modelar la proyección de los objetos del espacio tridimensional en los píxeles del plano imagen.

Proyección sobre el CCD

Primeramente estudiaremos la proyección central de puntos del espacio tridimensional \mathbb{R}^3 sobre el espacio euclídeo bidimensional \mathbb{R}^2 .

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \text{modelo de proyección} \rightarrow \begin{bmatrix} u \\ v \end{bmatrix}$$

Figura 3.5: Función del modelo de proyección.

Para simplificar el modelo, la óptica de la cámara se asimilará a una única lente simple (figura 3.6):

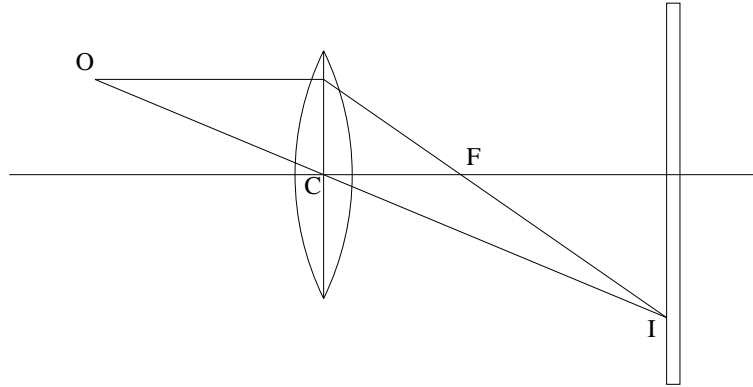


Figura 3.6: Modelo de proyección.

Suponiendo que la imagen está enfocada, entonces la distancia entre el foco F y la lente es fija, pudiendo asimilar la lente a un punto C llamado centro óptico de la lente.

Esta asunción nos permite expresar el modelo mediante una transformación **lineal proyectiva**, que mapea entre el espacio de puntos reales y el espacio de puntos del CCD.

De esta manera la **proyección de perspectiva** del punto M de coordenadas (x, y, z) sobre el CCD, m de coordenadas (u, v) vendrá definida por la intersección del plano CCD con la recta que une M con el centro óptico C (ver figura 3.7).

El sistema de referencia de la escena estará en el centro óptico de la lente. El modelo de proyección deberá calcular el punto del CCD en el que incide la recta que une M (expresado en el sistema de referencia de la escena) y pasa por el origen C, y expresarlo en el sistema de referencia del CCD.

De esta manera, aplicando semejanza de triángulos, obtenemos las expresiones 3.2 y 3.3 que relacionan un punto M(x,y,z) en el mundo real con su proyección m(u,v) en el CCD.

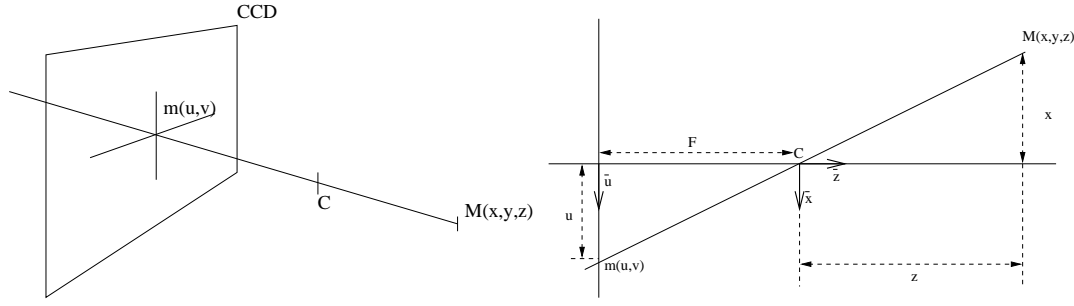


Figura 3.7: Esquema del modelo de proyección.

$$u = -f \frac{x}{z} \quad (3.2)$$

$$v = -f \frac{y}{z} \quad (3.3)$$

Proyección utilizando coordenadas homogéneas

Si observamos bien las ecuaciones 3.2 y 3.3 veremos que a cada punto del CCD no le corresponde un único punto del mundo real, sino que le corresponde un rayo óptico, es decir, le corresponden todos los puntos que pasan por el punto del CCD y por el centro óptico de la lente. Para poder expresar dichos rayos utilizamos las **coordenadas homogéneas**.

A partir de un punto en coordenadas cartesianas $x = [x, y]^T$, sus coordenadas homogéneas serán $x = [x, y, 1]^T$. Así, lo que antes era un punto, ahora representa una dirección, la que une dicho punto con el origen, que en nuestro caso es el centro de proyección de la escena, que está situado en el centro óptico. Así, si parametrizamos dicho punto en coordenadas homogéneas con un factor de escala λ de la forma $\lambda x = [\lambda x, \lambda y, \lambda]$ tendremos la familia de puntos que pertenecen a dicho rayo óptico.

Pasando las ecuaciones 3.2 y 3.3 a forma matricial y en coordenadas homogéneas, tenemos que:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fx/z \\ fy/z \end{bmatrix} \rightarrow \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.4)$$

Desplazamiento del punto principal: parámetros intrínsecos

Para pasar de una imagen física proyectada en el CCD, debemos pasar a una matriz de píxels (figura 3.8).

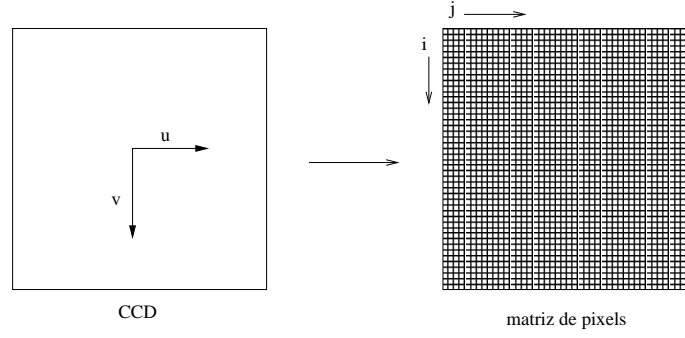


Figura 3.8: Paso del CCD a una matriz de píxeles en memoria

Podemos modelizar dicho cambio mediante un cambio de sistema de referencia:

$$j = k_u u + j_0 \quad (3.5)$$

$$i = k_v v + i_0 \quad (3.6)$$

donde:

- (i_0, j_0) es el punto principal de la imagen
- k_u es la longitud de una celda del CCD (pixels/u.longitud)
- k_v es la altura de una celda del CCD (pixels/u.longitud)

Así, en coordenadas homogéneas:

$$\begin{aligned}
 \lambda \begin{bmatrix} j \\ i \\ 1 \end{bmatrix} &= \begin{bmatrix} k_u & 0 & j_0 \\ 0 & k_v & i_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \\
 &= \begin{bmatrix} k_u & 0 & j_0 \\ 0 & k_v & i_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \\
 &= \begin{bmatrix} k_u f & 0 & j_0 & 0 \\ 0 & k_v f & i_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.7)
 \end{aligned}$$

Por último, haciendo los cambios de variable $\alpha_u = k_u f$ y $\alpha_v = k_v f$ e introduciendo un nuevo parámetro s , al que llamaremos sesgo, que modifica los vectores \bar{u} y \bar{v} para que no tengan que ser perpendiculares, nos queda:

$$\lambda \begin{bmatrix} j \\ i \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & s & j_0 & 0 \\ 0 & \alpha_v & i_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.8)$$

De esta manera queda definida la matriz de calibración de la cámara:

$$K = \begin{bmatrix} \alpha_u & s & j_0 \\ 0 & \alpha_v & i_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

compuesta por los **parámetros intrínsecos** de la cámara:

- α_u, α_v llamadas focales, son las distancias focales expresadas en pixels
- s llamado sesgo, cuando el ángulo que forman \bar{u} y \bar{v} es 90° , entonces $s = 0$
- (i_0, j_0) llamado punto principal, es la posición del plano del CCD en la que corta el eje axial

Rotación y translación de la cámara: parámetros extrínsecos

Normalmente nos interesará expresar las coordenadas de los objetos tridimensionales proyectados en el plano imagen respecto a un sistema de referencia externo a la cámara que hará el papel de “origen de coordenadas del mundo”.

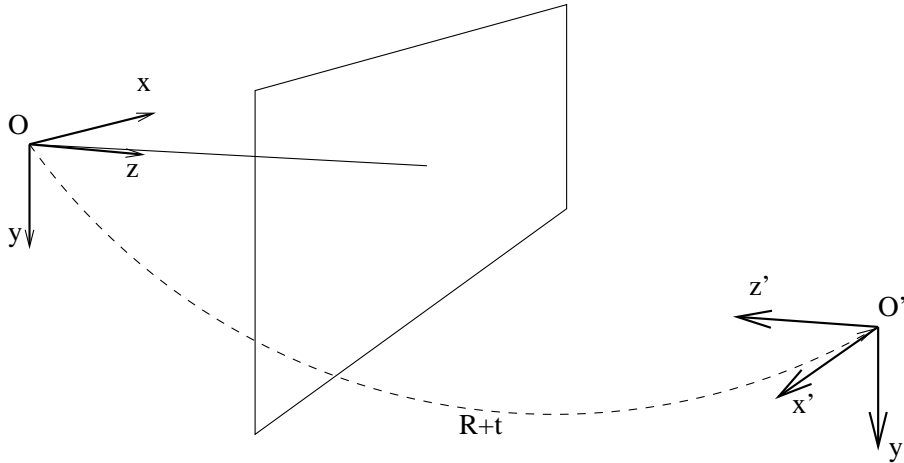


Figura 3.9: Cambio de sistema de referencia en el modelo extrínseco

Así, deberemos aplicar una rotación junto con una translación sobre el sistema de referencia que utilizábamos hasta ahora, es decir, multiplicar la expresión 3.8 por una matriz de transformación que sea composición de una rotación más una translación:

$$\begin{bmatrix} j \\ i \\ 1 \end{bmatrix} = K_{3 \times 3} [R_{3 \times 3} | \bar{t}_{3 \times 3}] \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \quad (3.10)$$

De esta manera quedan definidos los **parámetros extrínsecos**: $\bar{t}(t_x, t_y, t_z)$ y $R(\alpha, \beta, \gamma)$.

Así, suele definirse la matriz de proyección completa como:

$$P = K[R|t] \quad (3.11)$$

Geometría de escenas planas

Si la información que nos interesa extraer es la posición que ocupa un objeto sobre un plano conocido, los datos que necesitamos son menores, únicamente debemos construir una transformación lineal proyectiva, también llamada **homografía**. Una homografía es una transformación proyectiva que relaciona el espacio de puntos del plano imagen con el espacio de puntos del plano que conocemos, que en nuestro caso se tratará del plano del suelo $z = 0$. De esta manera, una homografía H será una matriz 3×3 de la forma:

$$\begin{bmatrix} j \\ i \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.12)$$

Supongamos que la ecuación del plano de la homografía es:

$$ax + by + cz + d = 0 \Leftrightarrow \left[\frac{-a}{d}, \frac{-b}{d}, \frac{-c}{d} \right] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 1 \quad (3.13)$$

Y llamando $\bar{n}^T = \left[\frac{-a}{d}, \frac{-b}{d}, \frac{-c}{d} \right]$:

$$\Pi : \bar{n}^T \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 1 \quad (3.14)$$

Ahora, sustituyendo en 3.10:

$$\begin{aligned}
\lambda \begin{bmatrix} j \\ i \\ 1 \end{bmatrix} &= K[R|\bar{t}] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = K[R|\bar{t}] \begin{bmatrix} x \\ y \\ z \\ \bar{n}^T \begin{bmatrix} x \\ y \\ z \end{bmatrix} \end{bmatrix} = \\
&= K[R|\bar{t}] \begin{bmatrix} I_{3 \times 3} \\ \bar{n}^T \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = [R + \bar{t}\bar{n}^T] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.15)
\end{aligned}$$

Por lo que obtenemos que:

$$H = K[R + \bar{t}\bar{n}^T] \quad (3.16)$$

Para calcular la homografía de un plano, supondremos que el sistema de referencia de la escena está sobre el plano, siendo así la ecuación de dicho plano es $z = 0$. De esta manera el problema se reduce a hallar una transformación proyectiva entre dos planos bidimensionales:

$$\lambda \begin{bmatrix} j \\ i \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.17)$$

Así, hallaremos un sistema en que las ecuaciones sean las componentes de H:

$$\begin{aligned}
\lambda \begin{bmatrix} j \\ i \\ 1 \end{bmatrix} &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \rightarrow \left\{ \begin{array}{l} \lambda j = h_{11}x + h_{12}y + h_{13} \\ \lambda i = h_{21}x + h_{22}y + h_{23} \\ \lambda = h_{31}x + h_{32}y + h_{33} \end{array} \right\} \rightarrow \\
&\rightarrow \left\{ \begin{array}{l} h_{11}x + h_{12}y + h_{13} - h_{31}jx - h_{32}jy - h_{33}j = 0 \\ h_{21}x + h_{22}y + h_{23} - h_{31}ix - h_{32}iy - h_{33}i = 0 \end{array} \right\} \quad (3.18)
\end{aligned}$$

De esta manera, tomaremos la mayor cantidad de puntos posibles sobre el plano y construiremos un sistema sobredimensionado que resolveremos mediante mínimos cuadrados, es decir, hallaremos el autovector asociado al menor autovalor de la matriz $B = A^T A$ donde A es la matriz del sistema de ecuaciones previamente citado.

3.2.3. Filtro de Kalman

En 1960, Rudof Emil Kalman publicó su famoso artículo *A New Approach to Linear Filtering and Prediction Problems* [9] donde describía una solución recursiva al problema de filtrado lineal de señales discretas [23, 6].

El filtro de Kalman en sí es un conjunto de ecuaciones matemáticas que nos permiten estimar el estado, actual o futuro, de un sistema contaminado por ruido gaussiano, es decir, ruido expresable mediante una distribución normal $N(\mu, C)$, de manera que se minimiza la media del error cuadrático.

Este filtro encaja dentro de la teoría de control moderna (teoría de espacio de estados) como un estimador robusto del estado del sistema a controlar, aunque en nuestro caso carecemos de entrada de control, únicamente deseamos estimar el estado del sistema.

La implementación de este tipo de filtro es muy similar a la de un algoritmo computacional iterativo, en el que en cada iteración se predice el estado del sistema en el siguiente instante futuro y posteriormente se corrige la estimación del estado del sistema haciendo uso de las mediciones (observaciones) que tomamos de él:

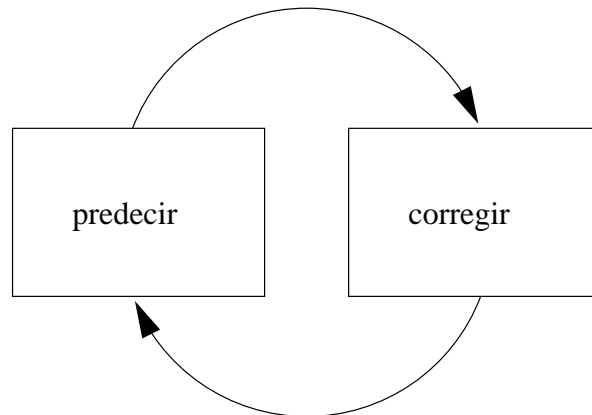


Figura 3.10: Esquema general del bucle iterativo del filtro de Kalman.

El proceso a estimar

El proceso cuyo estado $x \in \mathfrak{R}^n$ se desea estimar vendrá expresado mediante una ecuación en diferencias de la forma:

$$x_k = \Phi x_{k-1} + w_{k-1} \quad (3.19)$$

donde:

- x_k (n x 1) es el estado del sistema en el instante k
 Φ (n x n) es la matrix que relaciona x_{k-1} con x_k
 w_k (n x 1) es la variable aleatoria que representa el ruido que afecta al proceso

y las observaciones $z \in \Re^m$ de dicho estado vienen expresadas por:

$$z_k = Hx_k + v_k \quad (3.20)$$

donde:

- z_k (m x 1) es la observación del sistema en el instante k
 H (m x n) es la matriz que relaciona idealmente (en ausencia de ruido) la observación con el estado del proceso
 v_k (m x 1) es la variable aleatoria que representa el ruido que afecta a las observaciones

Como ya se mencionó previamente, el ruido que afecta tanto al estado del sistema como a las observaciones debe ser de naturaleza gaussiana, por lo que las variables aleatorias w_k y v_k deben seguir sendas normales de la forma:

$$p(w) \sim N(0, Q) \quad (3.21)$$

$$p(v) \sim N(0, R) \quad (3.22)$$

donde:

- Q (n x n) es la matriz de covarianza de $p(w)$
 P (m x m) es la matriz de covarianza de $p(v)$

En las ecuaciones 3.19,3.20,3.21,3.22 las matrices Φ, H, Q, R se asumen constantes a lo largo del tiempo a pesar de que en la realidad éste puede no ser el caso.

Orígen del filtro

Partiendo de que tenemos conocimiento del estado del sistema previo al instante k, definimos una estimación *a priori* del estado del sistema en dicho instante $\hat{x}_k^- \in \Re^n$. Así, el error cometido en dicha estimación es:

$$e_k^- = x_k - \hat{x}_k^- \quad (3.23)$$

y su matriz de covarianza asociada:

$$P_k^- = E[e_k^- e_k^{-T}] = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \quad (3.24)$$

Así, para actualizar dicha estimación *a priori* usaremos el error entre la observación tomada y la esperada, ponderado con un “factor de influencia”:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-) \quad (3.25)$$

donde:

- \hat{x}_k (n x 1) es la estimación actualizada
- K_k (n x m) es el factor de influencia (al que posteriormente llamaremos ganancia de Kalman)

El problema ahora consiste en hallar un factor de influencia K_k tal que la estimación actualizada sea óptima, es decir, que minimice el error cuadrático medio.

Para ello asociaremos primero una matriz de covarianza del error a la estimación actualizada (*a posteriori*):

$$P_k = E[e_k e_k^T] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \quad (3.26)$$

Ahora, sustituyendo 3.20 en 3.25 e introduciendo la expresión de \hat{x}_k obtenida en 3.26 obtenemos:

$$P_k = E\{[x_k - \hat{x}_k^-] - K_k(H_k x_k + v_k - H_k \hat{x}_k^-) \quad (3.27)$$

$$[(x_k - \hat{x}_k^-) - K_k(H_k x_k + v_k - H_k \hat{x}_k^-)]^T\}$$

Ya que $(x_k - \hat{x}_k^-)$ es la estimación *a priori* del error y que no está correlada con el error en la observación v_k , tenemos que:

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (3.28)$$

Nótese que para que K_k sea óptima debe minimizar los elementos individuales de la diagonal principal de P_k ya que éstos representan las varianzas en el error cometido en los elementos del vector de estado. Para minimizar la traza, teniendo que si se minimiza la traza se minimizan cada uno de los componentes de la diagonal principal, igualaremos la derivada de la traza respecto a K_k a cero.

Expandiendo 3.28:

$$P_k = P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k (H_k P_k^- H_k^T + R_k) K_k^T \quad (3.29)$$

Teniendo en cuenta que el segundo y tercer términos son lineales en K y el cuarto es cuadrático, aplicando fórmulas de cálculo diferencial ¹

$$\frac{d(\text{traza}(P_k))}{dK_k} = -2(H_k P_k^-)^T + 2K_k(H_k P_k^- H_k^T + R_k) \quad (3.33)$$

E igualando la derivada a cero y resolviendo, obtenemos una expresión de la ganancia óptima K_k :

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (3.34)$$

La K_k hallada (es decir, la que minimiza el error cuadrático medio) se llama *ganancia de kalman*.

La covarianza asociada a ella, partiendo de 3.34 y usando 3.29, es:

$$\begin{aligned} P_k &= (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T = \\ &= P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k (H_k P_k^- H_k^T + R_k) K_k^T = \\ &= P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- = \\ &= P_k^- - K_k (H_k P_k^- H_k^T + R_k) K_k^T = \\ &= (I - K_k H_k) P_k^- \end{aligned} \quad (3.35)$$

Ahora que ya tenemos una expresión de la ganancia óptima (3.35), ya podemos mediante 3.25 asimilar la observación a nuestra estimación del estado. En la primera iteración del filtro, necesitaremos unos \hat{x}_k^- y P_k^- iniciales, pero en iteraciones sucesivas hallaremos \hat{x}_{k+1}^- proyectando \hat{x}_k mediante la matriz de transición (3.19) (ignoraremos la contribución de w_k ya que la media del error es cero y no está correlada con los w 's previos)

$$\hat{x}_{k+1}^- = \Phi_k \hat{x}_k \quad (3.36)$$

Para calcular la matriz de covarianza del error asociado a \hat{x}_{k+1}^- primero hallamos la expresión del error *a priori*:

$$\frac{d[\text{traza}(AB)]}{dA} = B^T \text{ (AB debe ser cuadrada)} \quad (3.30)$$

$$\frac{d[\text{traza}(ACA^T)]}{dA} = 2AC \text{ (C debe ser simétrica)} \quad (3.31)$$

donde la derivada de un escalar respecto a una matriz se define como

$$\frac{ds}{dA} = \begin{bmatrix} \frac{ds}{da_{11}} & \frac{ds}{da_{12}} & \cdots \\ \frac{ds}{da_{21}} & \frac{ds}{da_{22}} & \cdots \\ \vdots & & \end{bmatrix} \quad (3.32)$$

$$\begin{aligned}
e_{k+1}^- &= x_{k+1} - \hat{x}_{k+1}^- = \\
&= (\Phi_k x_k + w_k) - \phi_k \hat{x}_k = \\
&= \Phi_k e_k + w_k
\end{aligned} \tag{3.37}$$

Y ahora, como w_k y e_k tiene correlación cruzada nula debido a que w_k es el ruido del proceso para el paso siguiente at_k , la expresión de P_{k+1}^- es:

$$\begin{aligned}
P_{k+1}^- &= E[e_{k+1}^- e_{k+1}^{T-}] = E[(\Phi_k e_k + w_k)(\Phi_k e_k + w_k)^T] = \\
&= \Phi_k P_k \Phi_k^T + Q_k
\end{aligned} \tag{3.38}$$

De esta manera se completa el ciclo iterativo del filtro de Kalman mediante las ecuaciones 3.25,3.34,3.35,3.36 y 3.38:

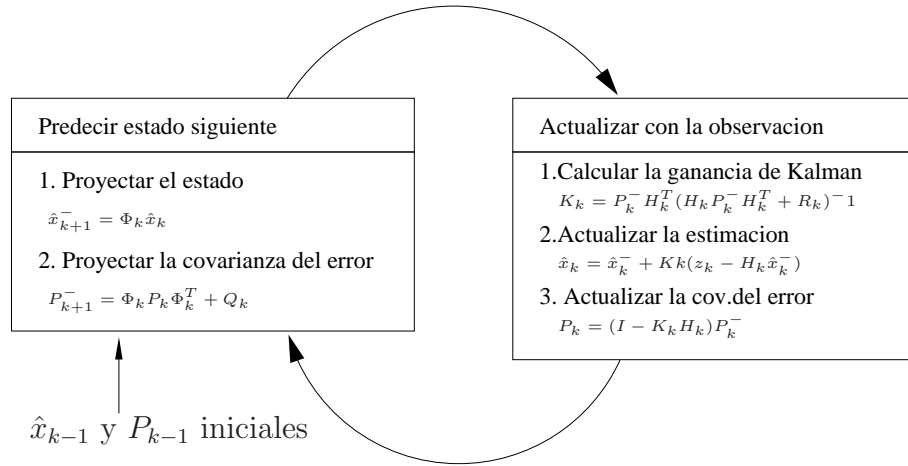


Figura 3.11: Bucle iterativo del filtro de Kalman.

Diseño de un filtro de Kalman apropiado

El diseño de un filtro de Kalman destinado a estimar el estado de un proceso puede dividirse en dos partes:

Modelo del proceso Lo que nos interesa modelar es el estado de una partícula que formará parte del coche; como “estado” de la partícula, tomaremos sus coordenadas cartesianas $\bar{u} = (x, y, z)$. Como modelo cinemático de la partícula, se eligió un **modelo de aceleración constante**[12], cuyas ecuaciones de movimiento son:

$$u_i = u_{i-1} + \Delta t v_{i-1} + \frac{1}{2}(\Delta t)^2 a_{i-1} + \epsilon_i \quad (3.39)$$

$$v_i = v_{i-1} + \Delta t a_{i-1} + \varsigma_i \quad (3.40)$$

$$a_i = a_{i-1} + \xi_i \quad (3.41)$$

Expresando 3.39, 3.40 y 3.41 de forma matricial:

$$\begin{bmatrix} u \\ v \\ a \end{bmatrix}_i = \begin{bmatrix} 1 & \Delta t & \frac{(\Delta t)^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ a \end{bmatrix}_{i-1} + \begin{bmatrix} \epsilon \\ \varsigma \\ \xi \end{bmatrix}_i \quad (3.42)$$

donde $[\epsilon, \varsigma, \xi]^T$ es el ruido en el proceso, que asumimos gaussiano.

De esta manera, las variaciones en la aceleración de los vehículos se asimilan como ruido en el proceso.

Como nuestro caso es bidimensional y los errores en las dos dimensiones en que trabajamos están correlados, debemos incorporar las ecuaciones para ambas dimensiones en una sola expresión matricial:

$$\begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix}_i = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \frac{(\Delta t)^2}{2} & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \frac{(\Delta t)^2}{2} \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix}_{i-1} + \bar{w}_i \quad (3.43)$$

donde \bar{w} es la variable aleatoria que representa al ruido que afecta al proceso.

Modelo de la observación del proceso La única observación que podemos tomar a cerca del estado del proceso es la posición aparente del vehículo, por lo que la relación entre el estado y la observación será:

$$\begin{bmatrix} x_{obs} \\ y_{obs} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} + \bar{v} \quad (3.44)$$

donde \bar{v} es la variable aleatoria que representa al ruido que afecta a la observación.

3.3. Modelo de conectividad

La detección y seguimiento de características, en nuestro caso esquinas, es sólo la base sobre la que se construye el algoritmo de tracking que en esta memoria se expone; dicha base carece de la carga semántica necesaria para la interpretación de los resultados a alto nivel: el problema que queremos resolver es el de seguir **vehículos**, no esquinas, siendo el método de agrupamiento de características el que nos permite salvar dicho salto.

La aproximación más directa para agrupar características, utilizada en [2, 3], hace uso de la información espacio-temporal de que disponemos sobre las características, resultado de su seguimiento por separado; de esta manera, se imponen restricciones que deben cumplir esquinas pertenecientes a un mismo vehículo, como distancia relativa constante y movimiento relativo cercano a cero; dichas restricciones serían altamente efectivas si lo que siguiésemos fuese la proyección vertical de las características, sin embargo lo que seguimos es la intersección del plano suelo con el rayo que une el centro óptico de la cámara con la esquina, lo que, tal y como se verá en el capítulo de experimentos, ocasiona que las “partículas” que seguimos vayan “expandiéndose” según se acercan a la línea del horizonte.

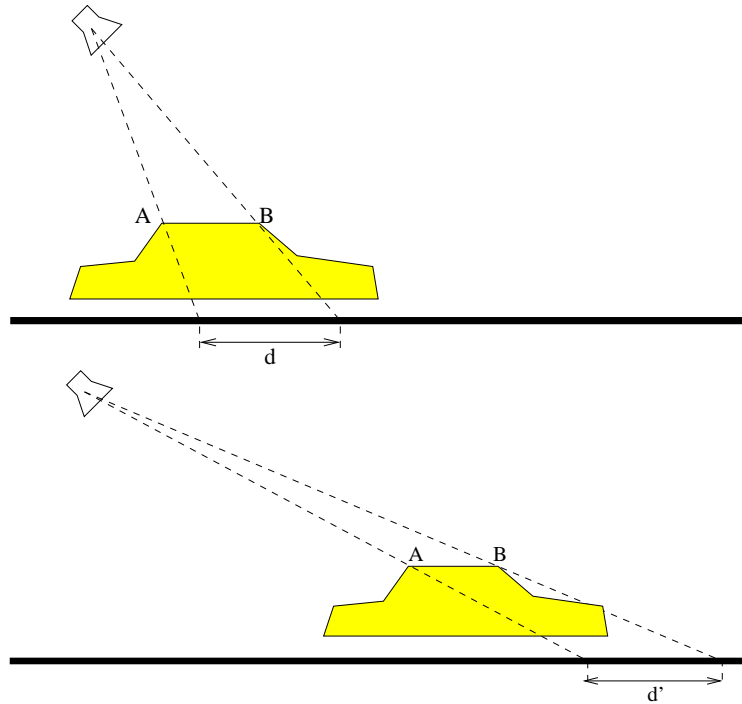


Figura 3.12: Desviación de las proyecciones de las esquinas.

Como vemos en la figura 3.12, la distancia que separa las proyecciones de las esquinas A y B en el primer fotograma, d será menor que su análoga en el segundo fotograma, d' , debido a que el ángulo de incidencia del rayo sobre el plano del suelo es menor

(se encuentran más próximas a la línea del horizonte).

Dicho inconveniente provoca que la eficacia de un algoritmo de agrupamiento de características de este tipo sea especialmente sensible a la localización y orientación de la cámara, produciendo resultados más correctos cuanto más cenital sea ésta.

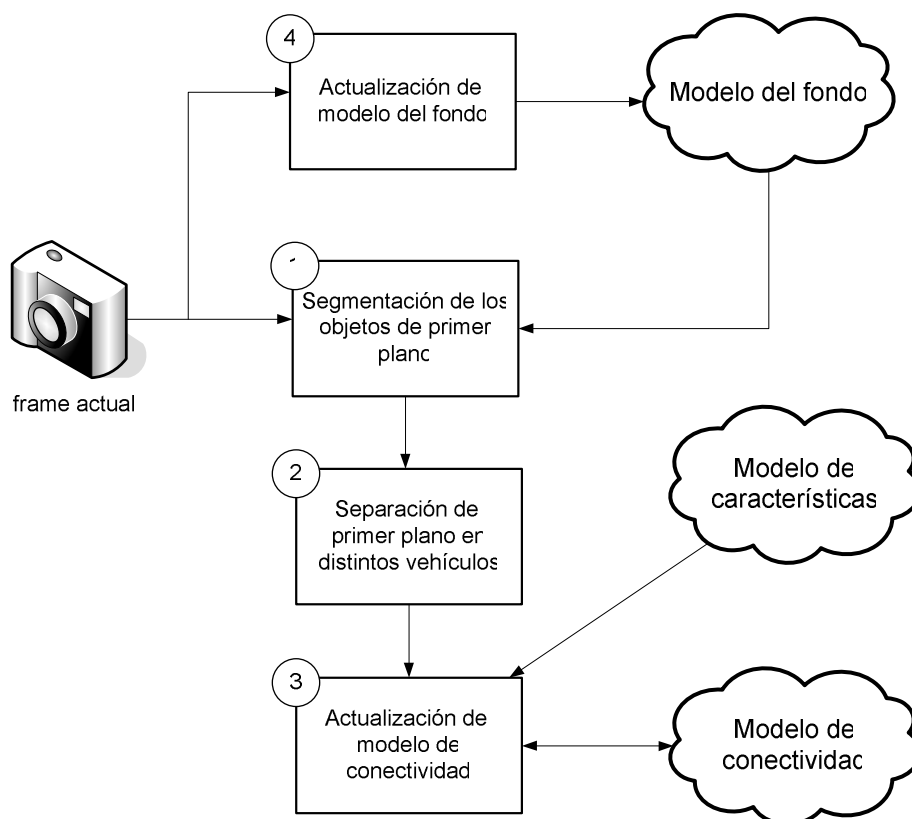


Figura 3.13: Actualización del modelo de conectividad.

Tras comprobar la ineficacia de este tipo de solución, se optó por una aproximación radicalmente distinta basada en la segmentación de los vehículos respecto del fondo mediante técnicas clásicas. Tal y como podemos observar de manera somera en la figura 3.13, el proceso que seguimos es:

1. Partiendo del fotograma actual (3.14) y un **modelo del fondo** (3.15) (en esta sección no detallaremos cómo se calcula el modelo del fondo, de ello se encarga la sección 3.3.1) se calcula su diferencia, de lo que obtenemos una imagen en la que “destacan” las diferencias entre el fondo y los elementos ajenos a él (3.16):
2. **Umbralizamos** dicha imagen, es decir, la transformamos en binaria (3.17), convirtiendo el valor de cada píxel en cero o en uno dependiendo si supera cierto umbral arbitrario elegido por nosotros (mediante ensayo-error). Ahora, nuestra imagen “diferencia” se ha transformado en una función booleana capaz de decirnos para cada píxel, si se trata de fondo o no.

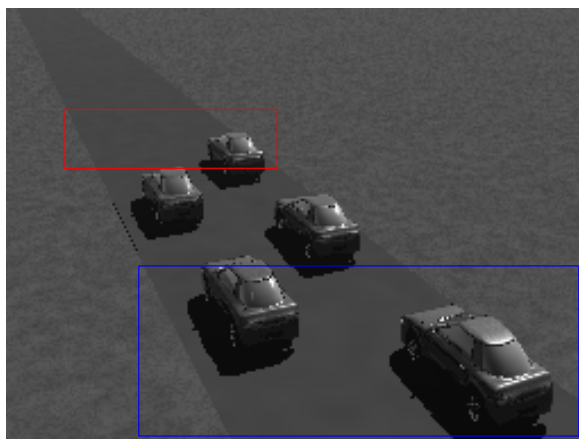


Figura 3.14: Fotograma actual.

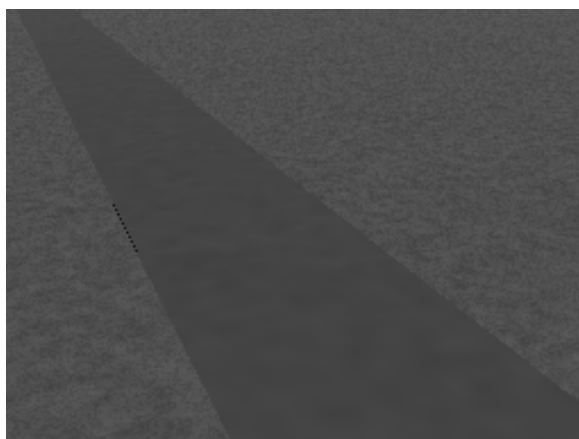


Figura 3.15: Modelo del fondo.

3. Separamos dicha imagen umbralizada en sus “**componentes conexas**” (3.18). Así, la función booleana que obtuvimos en el paso anterior se ha convertido en una función multivaluada que, para cada píxel, nos dice el vehículo en el que se encuentra.
4. Sabiendo la posición de cada esquina dentro de la imagen y usando la información que hemos obtenido como resultado del paso anterior, podemos saber

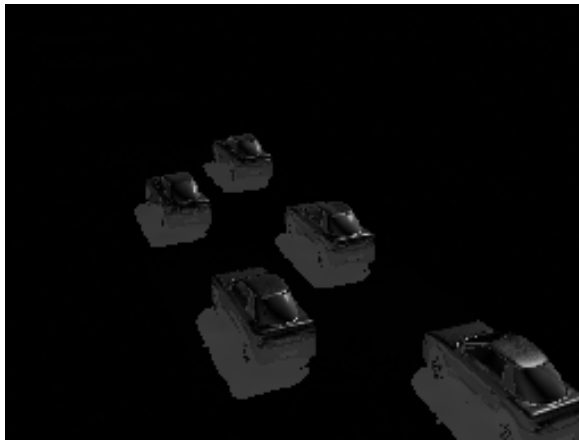


Figura 3.16: Diferencia del fotograma actual con el fondo.



Figura 3.17: Umbralización de la diferencia con el fondo.

si dos esquinas se encuentran “dentro” de un mismo vehículo (3.19). De esta manera, se construye un “grafo de conectividad” entre las esquinas que van encontrándose a lo largo de la secuencia: en cada fotograma, se determina para cada pareja de esquinas, si “pertenecen” al mismo vehículo; si es así, se le suma uno a la arista del grafo que las une; si no es así, se le resta uno.

5. Así, según van transcurriendo los fotogramas, cada pareja de esquinas va acu-

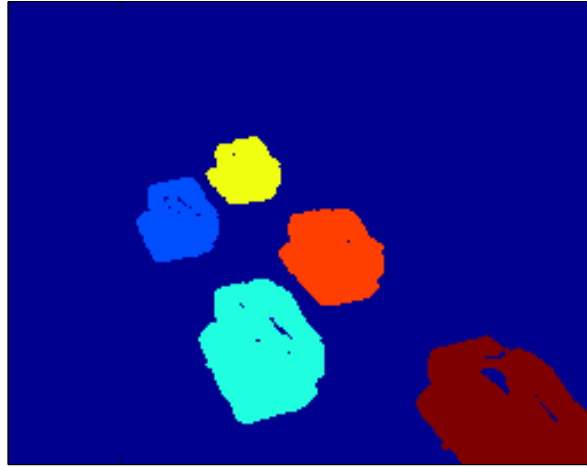


Figura 3.18: Componentes conexas en la umbralización de la resta con el fondo.

mulando votos (positivos o negativos), poblándose de esta manera el grafo de conectividad. Cuando alguna de las esquinas entra en la “ventana de salida”, se halla el conjunto de esquinas que están conectadas con ella (con peso mayor que cero) (3.20).

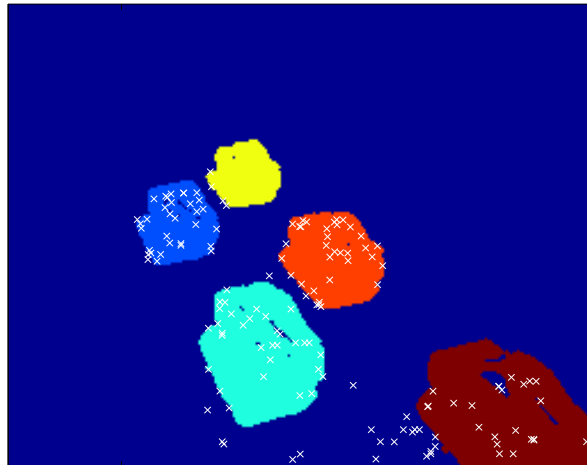


Figura 3.19: Agrupamiento de esquinas dentro de componentes conexas.

De esta manera conseguimos construir un grafo de conectividad entre las partículas del modelo de características, ponderado con los votos que cada conexión “va acumulando”, tal como podemos observar en la figura 3.20

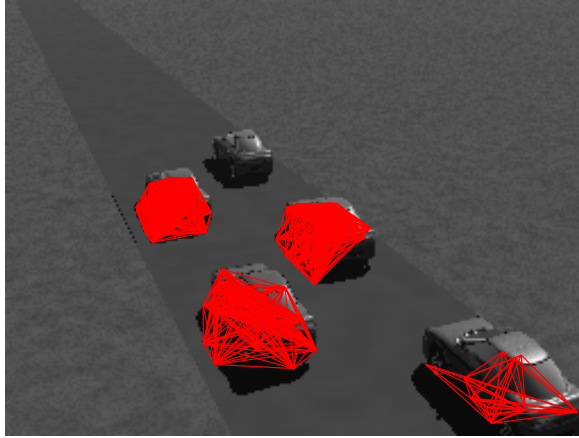


Figura 3.20: Grafo de conectividad entre las esquinas.

3.3.1. Modelo del fondo

Para poder realizar una correcta segmentación de los vehículos respecto del fondo, es imprescindible disponer de una imagen que refleje *el estado del fondo* en cada momento. Esto es de especial importancia en escenarios como el nuestro, en los que la cámara se encuentra en el exterior y por tanto sujeta a cambios de iluminación impredecibles debidos a múltiples motivos: avance de las horas (con el consiguiente *movimiento relativo del sol*), sombras de nubes, etc.

Para conseguirlo, partimos de una imagen estática del fondo en condiciones normales de luz, la cual iremos actualizando con cada nuevo fotograma, según la siguiente fórmula:

$$fondo_i = fotograma_i * \alpha + fondo_{i-1} * (1 - \alpha) \quad (3.45)$$

donde α es la *tasa de actualización* del fondo, es decir, la *velocidad* a la que el modelo se actualiza con el fotograma actual, y debe encontrarse entre 0 y 1. Darle un valor demasiado cercano a 0, hará que el modelo del fondo no se adapte con suficiente rapidez a los cambios de iluminación, mientras que un valor demasiado cercano a 1 hará que los vehículos que circulan *influyan* en el modelo. Dicho valor suele elegirse mediante ensayo-error y es altamente dependiente de las condiciones del escenario.

En la figura 3.21, podemos ver el modelo del fondo en dos momentos distintos de uno de los experimentos con secuencias reales. En dichas imágenes observamos cómo el algoritmo de actualización del modelo se ha adaptado a los cambios de iluminación, mientras que también ha evitado que los vehículos pasen a formar parte del modelo.



Figura 3.21: Modelo del fondo en dos instantes distintos de la secuencia real.

3.4. Extracción de vehículos

A partir de los modelos de características y conectividad tenemos toda la información necesaria para la extracción de la información cinemométrica que buscamos.

Para ello, definiremos una región rectangular en la imagen a la que denominaremos “región de salida”; cuando alguna esquina entre dentro de dicha región, buscaremos en el modelo de conectividad las partículas que están conectadas a ella (han ido acumulando suficientes “votos” a lo largo de la secuencia) y por tanto forman parte del mismo vehículo. Al “extraerse el vehículo”, eliminaremos toda la información de las partículas que lo componían, tanto del modelo de características tanto como del modelo de conectividad.

Capítulo 4

Resultados

Para evaluar el funcionamiento del trabajo realizado, se diseñaron dos tipos de experimentos:

- **Secuencias sintéticas:** se crearon secuencias generadas por computador mediante programas infográficos; al tener control total sobre los vehículos que aparecen en las secuencias, se han diseñado escenas que permiten observar el comportamiento del tracker ante diferentes condiciones.
- **Secuencias reales:** se obtuvieron imágenes procedentes de cámaras de tráfico reales, de manera que se pueda comprobar la validez de los métodos usados no sólo en secuencias sintéticas.

4.1. Experimentos con secuencias sintéticas

Sobre la generación de las escenas:

- El software infográfico utilizado para generar secuencias fue **povray** (The persistence of vision raytracer, www.povray.org), que utiliza la técnica del trazado de rayos para conseguir imágenes fotorrealistas. El método de descripción de escenas usado por povray es un lenguaje de programación con una sintaxis similar a C, permitiendo definir y situar los objetos en base a fórmulas matemáticas.
- Los elementos que conforman las escenas generadas (iluminación y vehículos) fueron tomados de la página personal de Jaime Vives Piqueres (www.ignorancia.org).

4.1.1. Objetivos

El objetivo de evaluar el sistema desarrollado con escenas generadas sintéticamente es **cuantificar** el desempeño del mismo, es decir, caracterizar su comportamiento con medidas numéricas que nos den información de la calidad de los resultados del sistema. Esto es posible debido al conocimiento **total** tanto de los elementos que conforman la escena como de su comportamiento dinámico a lo largo de la secuencia.

Las medidas que las escenas sintéticas nos permitirán averiguar son:

- Proporción de “capturas”: saber en qué grado el sistema es capaz de “detectar” a los vehículos.
- Robustez ante circunstancias “difíciles”, como vehículos con movimiento lateral, densidad de tráfico, etc.
- Exactitud de la estimación de la velocidad de los vehículos detectados.

4.1.2. Composición de las escenas

La escena sintética compuesta es sumamente sencilla, se reduce a un “desierto” por el que pasa una carretera sin marcas viales que tiene capacidad para dos carriles (véase figura 4.1). Todas las secuencias sintéticas fueron generadas simulando haber sido tomadas a 25 frames por segundo.



Figura 4.1: Escena sintética.

Con respecto a la situación y orientación de la cámara, se prepararon dos configuraciones distintas:

- **lateral**: simula una cámara de tráfico colocada en lo alto de un mástil anexo a la carretera. Estará colocada en $X = -8, Y = -18, Z = 8$ y estará mirando al punto $X = 2, Y = 5, Z = 0$.
- **cenital**: simula una cámara de tráfico colocada detrás de un panel luminoso, es decir, justo encima de la carretera. Estará colocada en $X = 0, Y = -18, Z = 8$ y estará mirando al punto $X = 2, Y = 5, Z = 0$.

En la figura 4.2 podemos observar cómo se ve la escena desde las dos localizaciones establecidas, lateral (izquierda) y cenital (derecha). En dicha figura se han añadido unos ejes ficticios para indicar el sistema de referencia usado (la longitud de los ejes es de 3 metros).

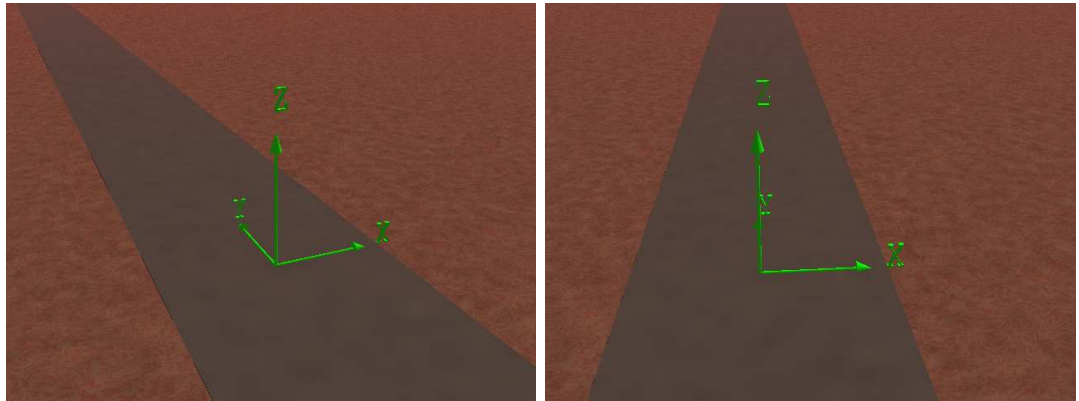


Figura 4.2: Escena sintética desde las localizaciones de la lateral y cenital.

4.1.3. Calibración del plano del suelo

Debido al control total ejercido sobre los elementos que conforman nuestra escena sintética, el proceso de calibración del plano del suelo se simplificó considerablemente: para lograrlo, se eliminó la carretera y se sustituyó la textura del suelo por una de cuadros negros y blancos, a modo de plantilla de calibración. Para poder tomar el mismo sistema de referencia en todas las secuencias, se colocaron cuatro esferas como guía.

De esta manera, tan sólo hubo que emparejar puntos del plano del suelo con puntos del plano imagen para cada posición distinta de la cámara y calcular las homografías correspondientes.

En la figura 4.3 podemos ver las plantillas de calibración para las dos localizaciones de cámara.

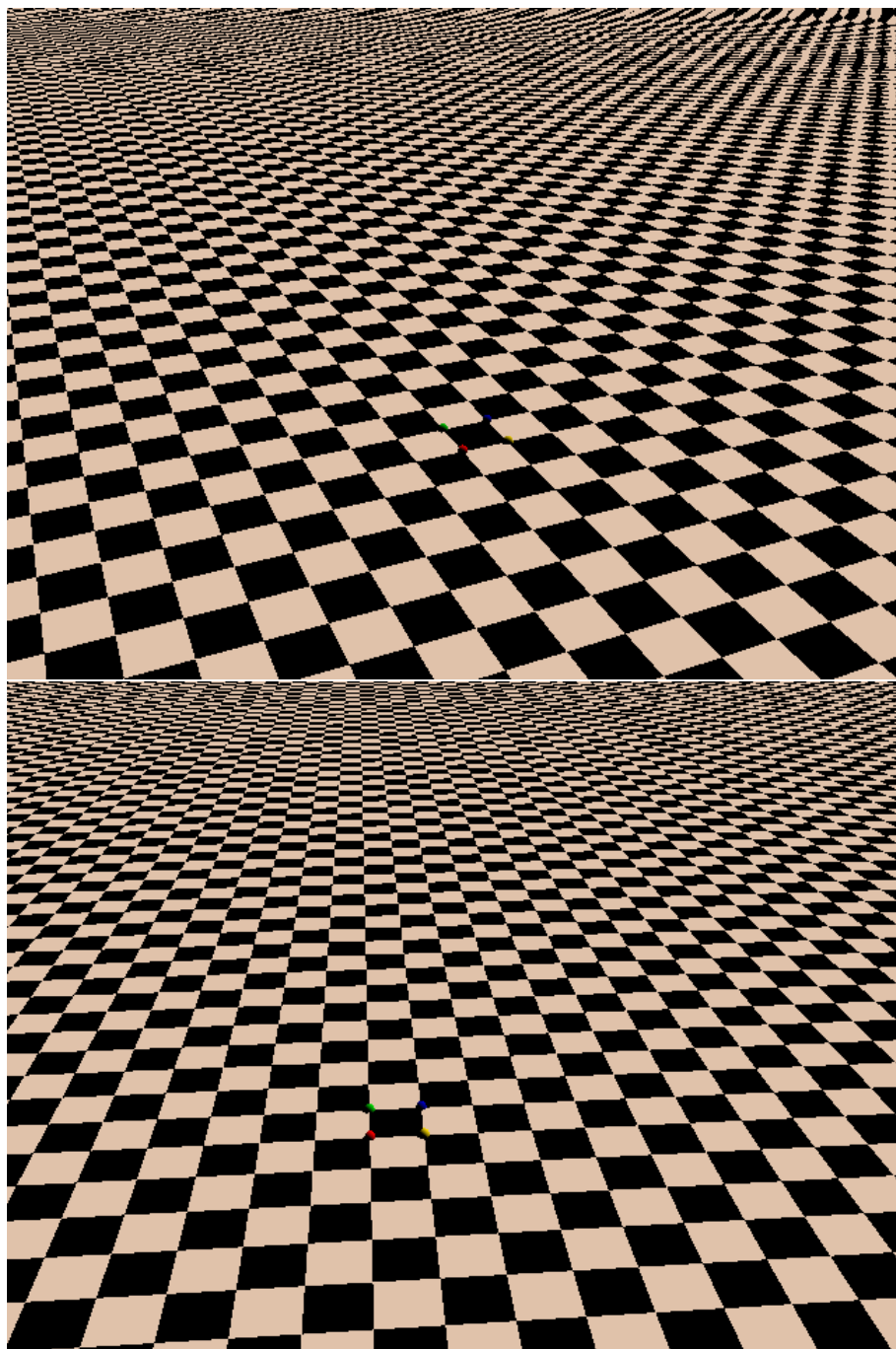


Figura 4.3: Plantillas de calibración de la escena sintética.

4.1.4. Experimentos

Seguimiento de vehículos a distintas velocidades en línea recta

En esta secuencia (4.4) circulan 3 convoys de 10 vehículos cada uno, circulando uno detrás de otro en línea recta; los 10 primeros van a 150 km/h, los 10 segundos a 100 km/h y los 10 terceros a 50 km/h.

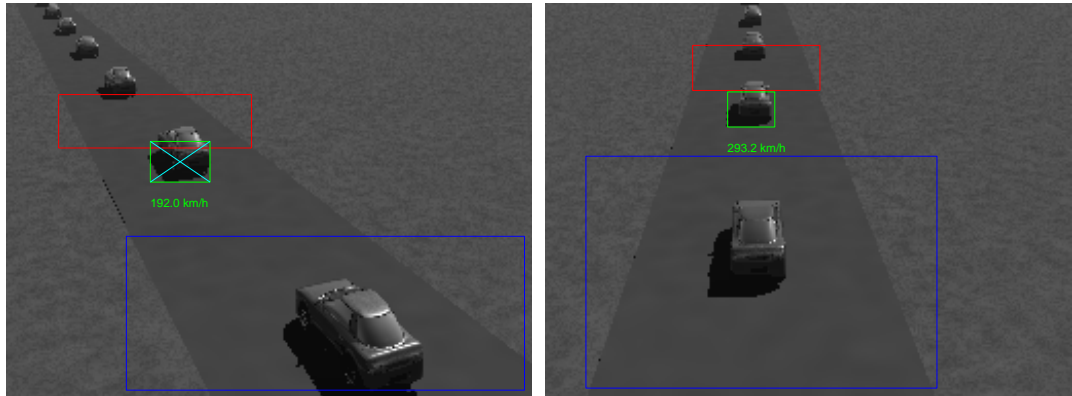


Figura 4.4: Snapshot del tracking de la secuencia del test 1.

El desempeño del tracker en las configuraciones lateral y cenital fue:

- Cámara lateral: el tracker fue capaz de captar un **100 %** de los vehículos. En la figura 4.5 podemos ver las velocidades medidas para los mencionados vehículos.

En el primer grupo de vehículos (que circulaba a 150 km/h), la media del error absoluto cometido fue 21,9 km/h (14,6 %), en el segundo grupo (a 100 km/h) fue de 10,6 km/h (10,6 %) y en el tercero (a 50 km/h) de 7,9 km/h (15,9 %).

- Cámara cenital: el tracker fue capaz de captar un **100 %** de los vehículos. En la figura 4.6 pueden apreciarse las velocidades medidas.

En el primer grupo de vehículos (150 km/h), la media del error absoluto cometido fue 6,7 km/h (4,5 %), en el segundo grupo (100 km/h) fue de 2,1 km/h (2,1 %) y en el tercero (50 km/h) de 0,8 km/h (1,6 %).

Vemos que el error cometido en las mediciones de velocidad es significativamente superior en el caso de la cámara lateral que en el de la cenital. Asimismo, vemos que el error es creciente con la velocidad del vehículo.

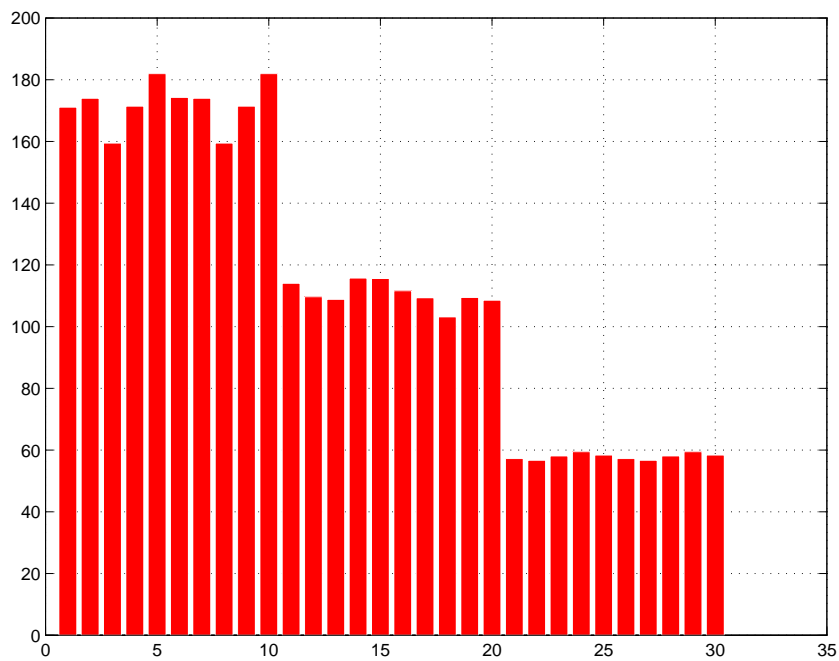


Figura 4.5: Velocidades medidas en el test 1, vista lateral.

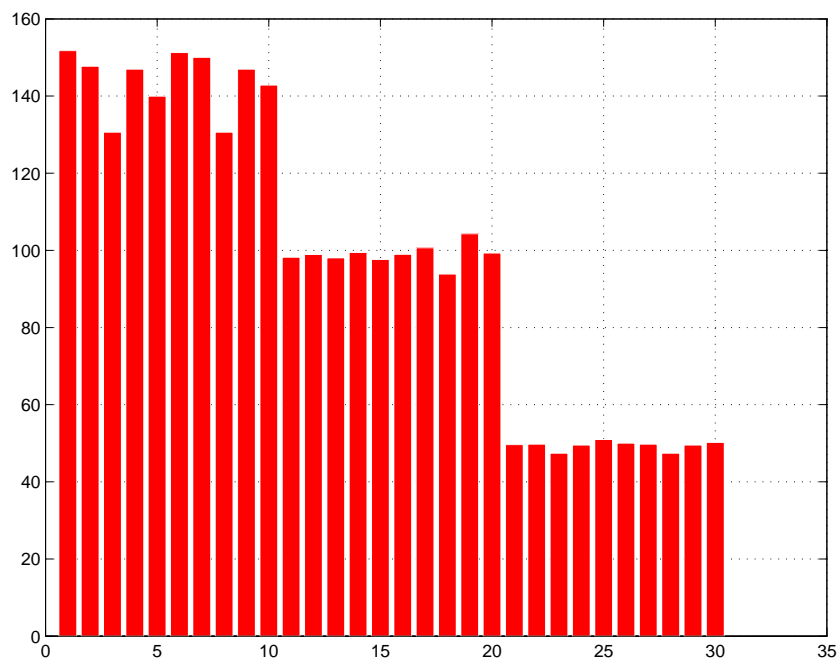


Figura 4.6: Velocidades medidas en el test 1, vista cenital.

Seguimiento de vehículos con movimiento lateral

Esta secuencia (4.7) es idéntica a la anterior salvo por que los vehículos, además de su movimiento hacia delante presentan desplazamiento lateral descrito por una senoide (van “haciendo eses”).

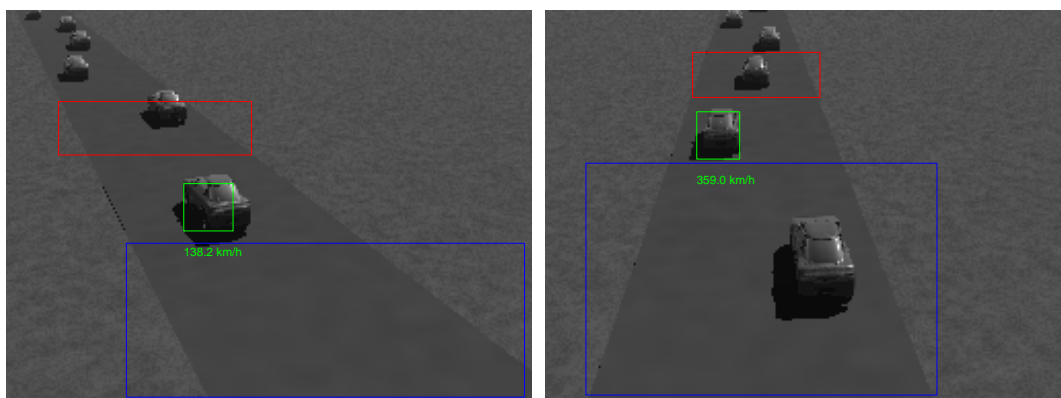


Figura 4.7: Snapshot del tracking de la secuencia del test 2.

El desempeño del tracker en las configuraciones lateral y cenital fue:

- Cámara lateral: el tracker fue capaz de captar un **93 %** de los vehículos. En la figura 4.8 podemos apreciar las medidas de velocidad tomadas.

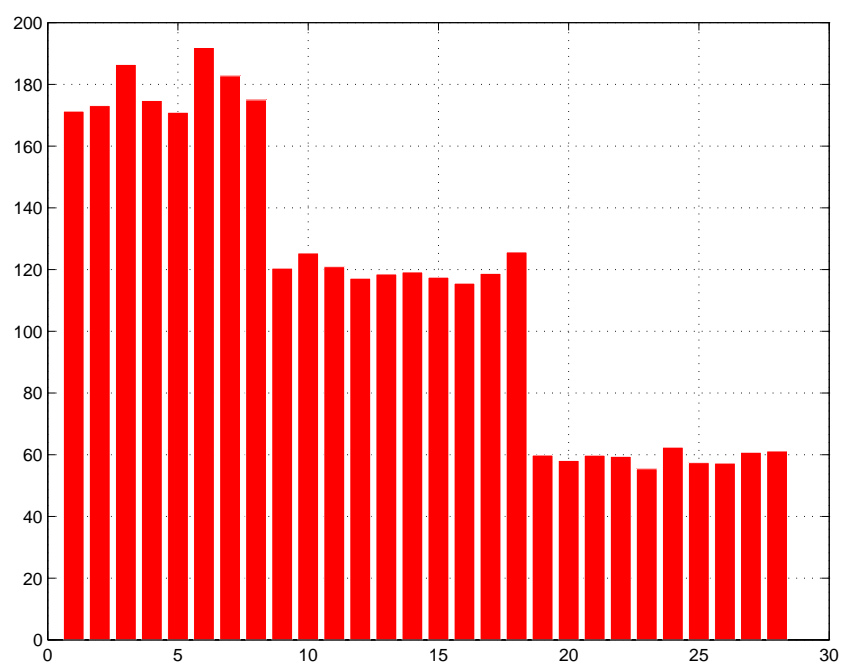


Figura 4.8: Velocidades medidas en el test 2, vista lateral.

En el primer grupo de vehículos (150 km/h), la media del error absoluto cometido fue 28,3 km/h (18,9%), en el segundo grupo (100 km/h) fue de 19,9 km/h (19,9%) y en el tercero (50 km/h) de 9,2 km/h (18,4%).

- Cámara cenital: el tracker fue capaz de captar un **100 %** de los vehículos. En la figura 4.9 podemos ver las velocidades medidas.

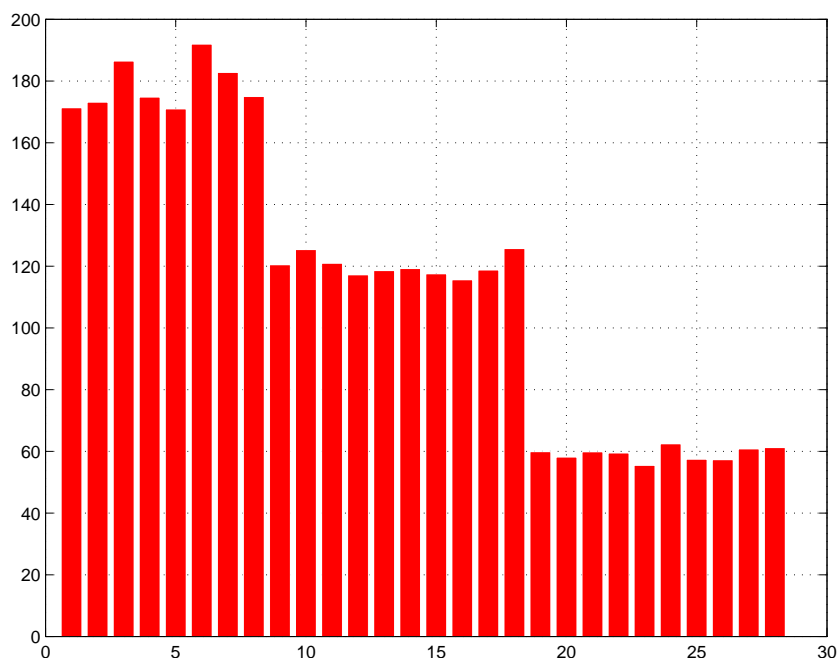


Figura 4.9: Velocidades medidas en el test 2, vista cenital.

En el primer grupo de vehículos (150 km/h), la media del error absoluto cometido fue 4,6 km/h (3,1%), en el segundo grupo (100 km/h) fue de 1,3 km/h (1,3%) y en el tercero (50 km/h) de 1,4 km/h (2,7%).

De este experimento puede extraerse las mismas conclusiones que en el anterior: el desempeño en la configuración con cámara cenital es muy superior al de la lateral y el error cometido crece con la velocidad a la que circula el vehículo. También comprobamos que el movimiento lateral de los coches también empeora la calidad de las medidas, llegando a hacer perder dos vehículos del grupo de los que avanza a 150 km/h.

Seguimiento con densidad de tráfico

En esta secuencia (4.10) aparecen dos hileras de coches, todos circulando a velocidad constante de 100 km/h.

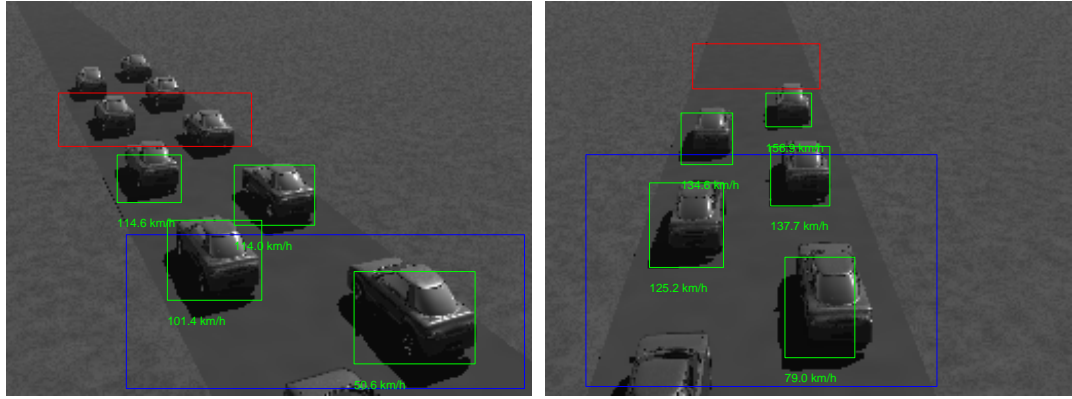


Figura 4.10: Snapshot del tracking de la secuencia del test 3.

El desempeño del tracker en las configuraciones lateral y cenital fue:

- Cámara lateral: el tracker fue capaz de captar el 100 % de los vehículos, la velocidad estimada media fue 114 km/h, con una desviación típica de 4.1 km/h (véase figura 4.11)

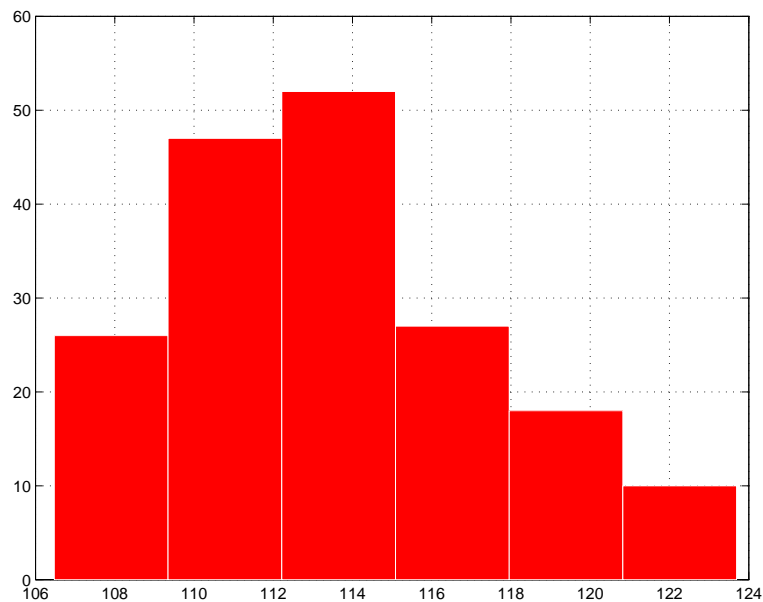


Figura 4.11: Histograma de las estimaciones de velocidad de los vehículos.

- Cámara cenital: el tracker fue capaz de captar el 100 % de los vehículos, la velocidad estimada media fue 98,6 km/h, con una desviación típica de 3 km/h (véase figura 4.11).

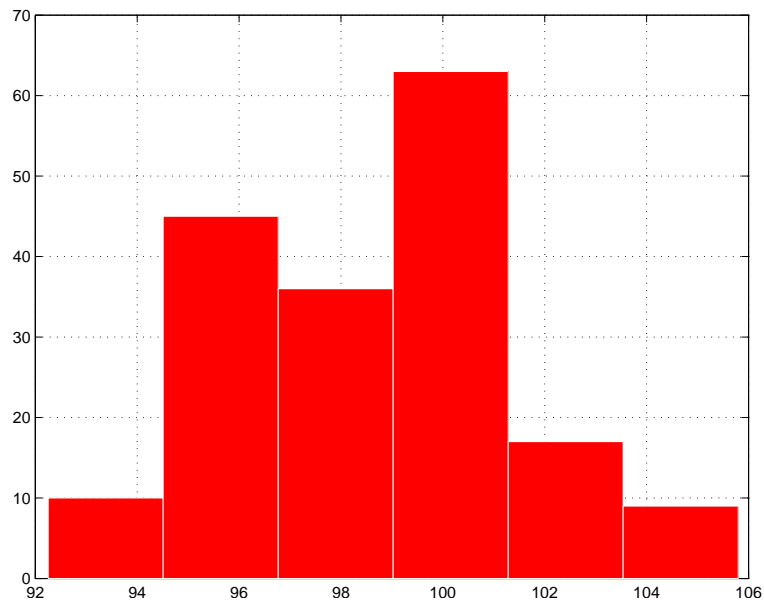


Figura 4.12: Histograma de las estimaciones de velocidad de los vehículos.

4.2. Experimentos con secuencias reales

Las secuencias de tráfico real de las que se disponía proceden de las cámaras de los túneles de Vallvidrera, a la salida de Barcelona. Las grabaciones, que datan del año 1992, estaban originalmente en soporte VHS, por lo que para poder ser tratadas fueron digitalizadas mediante una tarjeta capturadora de vídeo.

4.2.1. Objetivos

El objetivo de evaluar el sistema desarrollado con escenas reales es comprobar si sólo es válido para secuencias sintéticas o es realmente eficaz para imágenes del mundo real, tanto respecto a “captura” del vehículo como a medida de su velocidad. A diferencia de las pruebas con secuencias sintéticas, se desconoce los elementos que conforman las secuencias, por lo que no podremos “medir” la eficacia del algoritmo. Así mismo, que el sistema tenga un cierto desempeño para esta secuencia no es algo extrapolable a otras secuencias reales, ya que la variabilidad de las condiciones en el mundo real, no lo hace posible.

4.2.2. Composición de las escenas

Las secuencias de tráfico real corresponden a una cámara colocada en lo alto de un mástil anexo a la carretera apuntando a un fragmento de la carretera con dos carriles para un sentido y uno para el contrario, siendo los carriles de 3,5 metros de anchura (medida estándar).

4.2.3. Calibración del plano del suelo

El método normal para el calibrado del plano del suelo en condiciones normales sería desplazarse al lugar enfocado por las cámaras y hallar las coordenadas globales (mediante GPS o GPS diferencial) de algunos puntos sobre la carretera y posteriormente emparejarlos con sus correspondencias en el plano imagen, obteniendo así una homografía. Sin embargo, dado que no se dispuso de acceso físico al lugar apuntado por las cámaras, el calibrado del suelo tuvo que realizarse “a mano”, es decir, asumiendo distancias estándar (anchura de carriles y longitud media de los vehículos) elaborar un conjunto de correspondencias (en principio, cuanto mayor sea su número, menor será el impacto del error cometido). Así, y utilizando las líneas sobre el pavimento, se trazaron líneas que sirvieron como guía (véase figura 4.13); ésto unido a las “suposiciones” de que las líneas discontinuas del suelo median 3 metros continuos más 1 metro en blanco permitió obtener una homografía aproximada del plano del suelo.

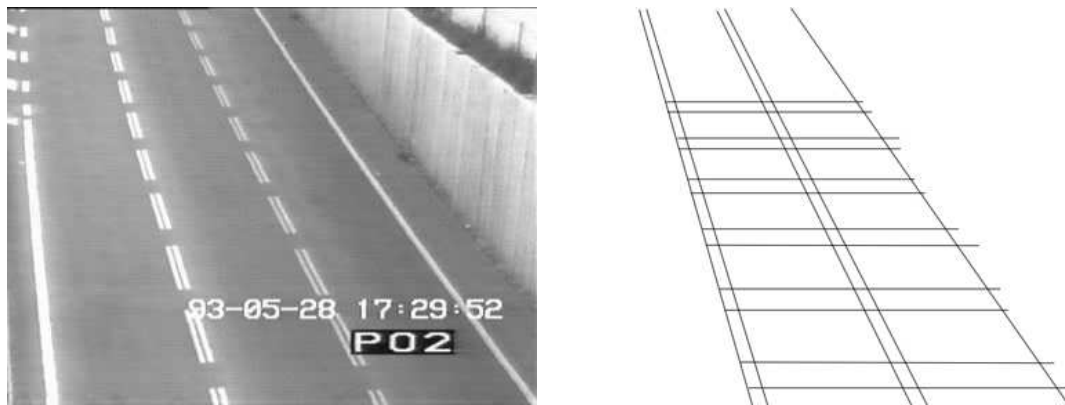


Figura 4.13: Calibración “a mano” de las escenas reales.

4.2.4. Experimentos

El tracker fue capaz de captar un **88 %** de los vehículos. En la figura 4.14 podemos algunos fotogramas de las capturas del tracker.

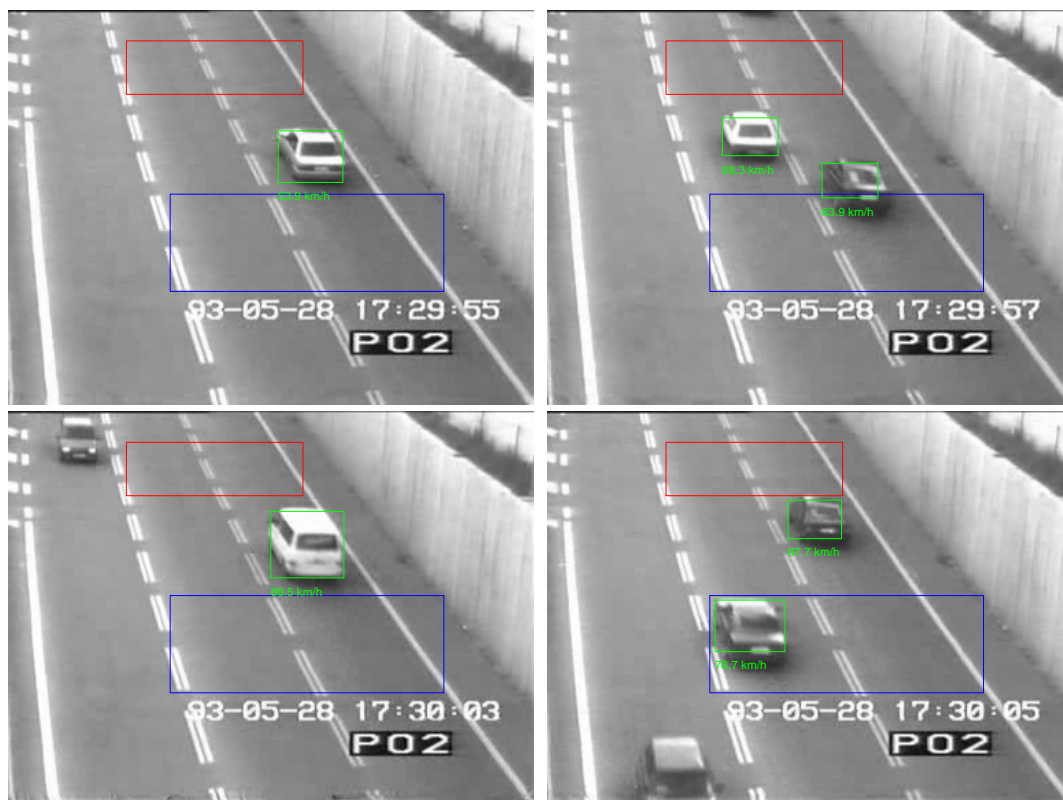


Figura 4.14: Fotogramas de la secuencia real.

Todos los fallos de captura del tracker se debieron a:

- Incorrecta segmentación: el algoritmo de segmentación de los coches respecto al fondo agrupó dos vehículos que estaban próximos. En la figura 4.15 podemos ver cómo el algoritmo agrupa incorrectamente varios vehículos.
- Poco contraste del vehículo con el fondo: el vehículo era de una tonalidad notablemente similar al fondo, por lo que el algoritmo de segmentación lo asimilaba a él.

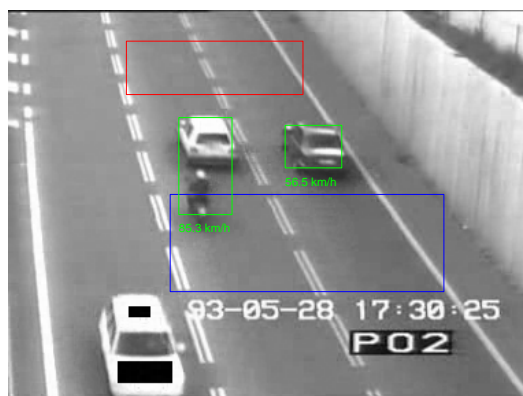


Figura 4.15: Error en la segmentación de vehículos en la secuencia real.

En la secuencia de prueba se producen constantes cambios de luz que dificultan la segmentación de los vehículos respecto al fondo. El modelo que el algoritmo mantiene del fondo permite una correcta segmentación; en la figura 4.16 podemos ver lo que el algoritmo “piensa” que es el fondo en dos instantes distintos de la secuencia.



Figura 4.16: Modelo del fondo en dos instantes distintos de la secuencia real.

4.3. Análisis de los resultados

4.3.1. Captura

De los resultados de los experimentos, sobre todo del primer y segundo experimentos con secuencias sintéticas, vemos que el grado de acierto en la “captura” de vehículos depende tres factores:

- Disposición de la cámara.
- su velocidad dentro de la ventana de extracción de características.
- la tasa de fotogramas por segundo captados por la cámara.

Respecto a la disposición de la cámara, la captura se ve afectada sobre todo ante movimientos laterales utilizando una configuración de cámara lateral.

Respecto al segundo y tercer factores, pueden concentrarse en uno solo: la distancia en píxels de una misma característica entre dos fotogramas sucesivos; si dicha distancia supera el área de búsqueda de encajes, la esquina nunca podrá ser encontrada en dos frames sucesivos y por tanto, al salir del área de extracción de características, se perderá. El aumentar el tamaño de dicha ventana de búsqueda tiene varios efectos: aumenta el tiempo de procesamiento de la secuencia, aumenta la probabilidad de “encontrar” la misma característica y aumenta la probabilidad de “falsos positivos” en el matching de características entre fotogramas sucesivos.

4.3.2. Agrupamiento de características

El correcto agrupamiento de características depende únicamente de la eficacia del algoritmo de segmentación de coches que en nuestro caso se trata de la diferencia entre el fotograma actual y el modelo del fondo que mantenemos. Dicho algoritmo se ve influenciado por:

- El contraste de los vehículos con el fondo: vehículos de tonalidad y luminosidad similares a los del fondo son candidatos a “mimetizarse” con él y por tanto a no ser captados por el tracker.
- Des-sincronización entre el modelo del fondo y el fondo real; cambios en la iluminación y elementos dinámicos del fondo, como hojas de árboles, degradan la calidad de la segmentación.

En las secuencias de tráfico real se daban continuos cambios de luz con los que el modelo probabilístico del fondo pudo lidiar sin problema. El caso de fondo dinámico, sería mucho más problemático.

4.3.3. Estimación de la velocidad de los vehículos

En las gráficas de los experimentos con secuencias sintéticas queda patente la deficiente estimación de la velocidad de los vehículos; el error cometido es relativamente bajo, pero se ve afectado en gran medida por la disposición de la cámara: en una configuración con cámara lateral, las medidas son mucho más inexactas, hecho que tiene origen en la propia naturaleza del tracker: lo que en realidad se “sigue” son las proyecciones en el plano del suelo de las características del vehículo; dichas proyecciones no se mueven a la misma velocidad que las características a las que corresponden, sino que aceleran según se aproximan a la línea del horizonte, tal y como podemos comprobar en la figura 4.17: en ella podemos ver la proyección en el plano del suelo de las características de un vehículo, así como la silueta que conforman, a lo largo de su aparición en la secuencia de la cámara lateral; dicha silueta va “estirándose” según se va alejando de la cámara. Dicho efecto se hace más patente en el caso de la cámara lateral, mientras que se ve atenuado cuanto más se acerca dicha disposición a una totalmente cenital.

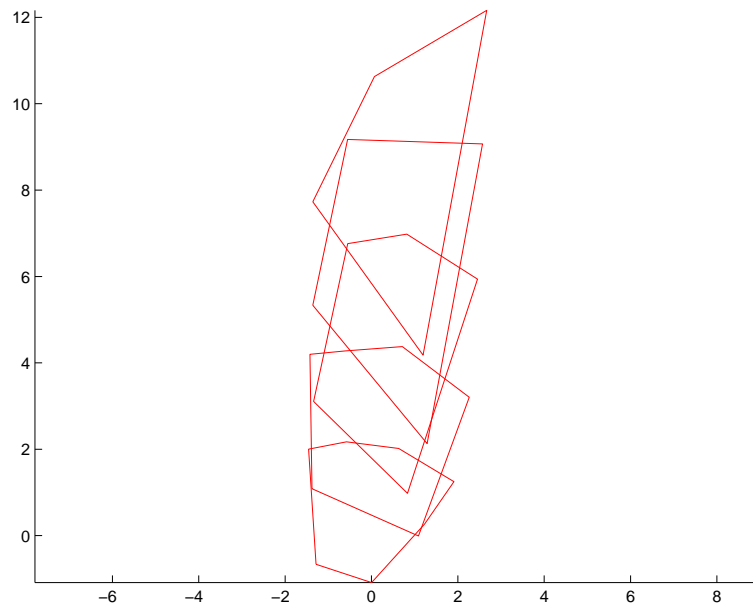


Figura 4.17: Proyección de la silueta del vehículo en el plano del suelo.

Capítulo 5

Conclusiones y líneas futuras

5.1. Conclusiones

En el presente proyecto se ha explorado y desarrollado un algoritmo capaz de medir la velocidad del tráfico rodado. Para conseguirlo, se implementó un *tracker* basado en puntos característicos, unido a un mecanismo de segmentación clásico basado en un modelo dinámico del fondo, lo que nos permite agrupar características en grupos de *más alto nivel*, es decir, los vehículos. Dada su naturaleza, algunas características positivas de dicho algoritmo son:

- Capacidad para seguir todo tipo de vehículos, sin necesidad de ningún conocimiento *a priori* sobre ellos.
- Robustez ante occlusiones parciales de los vehículos.
- Robustez a cambios de iluminación.

Aunque también presenta características no tan ventajosas:

- Necesidad de conocimiento *a priori* de la escena donde se va a situar la cámara para poder establecer correspondencias entre imágenes tomadas y el mundo real.
- Errores en la segmentación debido a la proximidad de vehículos entre sí.

Tal como se comentó en los capítulos introductorios, las soluciones para medición de la velocidad del tráfico rodado en la actualidad están basadas en emisores/receptores de trenes de pulsos de ondas de radio que permiten, gracias al efecto *doppler* y de manera instantánea, conocer la velocidad de los objetos que están pasando por

delante de dicha onda. Además, se necesita una cámara para tomar una fotografía al hipotético infractor en caso de que éste supere la velocidad máxima permitida.

También existen otras soluciones más baratas, aunque de menor precisión, como los bucles de inducción situados por debajo del firme. Cuando un vehículo circula por encima de ellos, se induce una corriente que permite aproximar su velocidad. Sin embargo este tipo de técnicas son altamente intrusivas, necesitando adaptar la carretera al dispositivo medidor.

Nuestra aproximación, carece de los problemas anteriormente mencionados y de los que adolecen las estrategias usadas hoy en día:

- Únicamente es necesaria una cámara y una unidad de procesamiento, nada más. El coste del equipo extra se elimina gracias a la ganancia en complejidad de la lógica del proceso.
- Es completamente inocuo a las infraestructuras sobre las que se vaya a instalar, no teniendo que ser modificadas en manera alguna.

Por tanto, la solución expuesta en la presente memoria parece una alternativa viable frente a las existentes actualmente en el mercado desde el punto de vista técnico.

5.2. Líneas futuras

5.2.1. Implementación *realtime*

La implementación del algoritmo expuesto se realizó en matlab debido a la agilidad de desarrollo de prototipos que presenta dicho entorno. Matlab es especialmente adecuado para los algoritmos que básicamente realizan operaciones con matrices de gran tamaño. Dado que nuestra aproximación tiene una naturaleza más procedimental, matlab no es especialmente apropiado desde el punto de vista de *performance*. Ello hace que nuestra implementación diste considerablemente de poder ser usada en tiempo real, es decir, el número de frames por segundo que es capaz de procesar es considerablemente inferior a la generada por una cámara de vídeo.

De esta manera, una posible línea de expansión sería el portar este algoritmo a un lenguaje de más *bajo nivel*, como C o C++, más apropiados para este tipo de estructura de procesamiento. Además, la existencia en el mercado de librerías como OpenCV, supliría la enorme cantidad de funcionalidad aprovechada del entorno matlab en la versión actual.

Asimismo, para conseguir una versión completamente *realtime* debería poder acortarse el tiempo de procesamiento de un frame, es decir, debería poder estimarse cuál es el tiempo máximo que puede tardarse en procesar un fotograma. En la versión

implementada para el presente proyecto esto no es posible; la razón es que el tiempo de procesamiento de un frame va en función de la cantidad de esquinas a seguir y agrupar, y ello viene únicamente determinado por la secuencia de vídeo de entrada. Así, para poder conseguir acortar el tiempo de procesamiento por frame, debería limitarse en número máximo de estinas que se están procesando en un determinado momento.

5.2.2. Mejora de la segmentación

El algoritmo de segmentación de los vehículos respecto del fondo utilizado es bastante primitivo, adoleciendo de problemas claros como los mostrados en la figura 4.15. Los problemas de segmentación actuales se ven aumentados según se aproximan los vehículos a la línea del horizonte, ya que el área que ocupan en la imagen es cada vez menor, de igual manera que el área ocupada por el espacio libre entre ellos

5.2.3. Autocalibración

Uno de los *handicaps* que presenta nuestra aproximación es la necesidad de conocimiento *a priori* acerca de la composición de la escena respecto a la cámara que toma las imágenes, es decir, la imposición de pasar por un proceso de calibración del plano del suelo para así poder obtener una homografía (véase sección 3.2.2).

Existen técnicas englobadas bajo el nombre de “autocalibración” ([25, 24, 20]) que permiten obtener conocimiento acerca de la escena frente a la que se encuentra la cámara sin tener ningún conocimiento *a priori* sobre ella. Dichas técnicas hacen uso de elementos de la imagen como líneas de fuga o la línea del horizonte para estimar los parámetros intrínsecos y extrínsecos del modelo pinhole de la cámara.

Las técnicas mencionadas, de ser añadidas a nuestra solución, permitirían prescindir del paso de calibración del plano del suelo, otorgándole mayor flexibilidad frente a entornos desconocidos o incluso cambiantes.

5.2.4. *Structure from motion*

De manera similar a la visión estéreo, que obtiene información tridimensional a partir de varias vistas bidimensionales de la misma escena, existen algoritmos, catalogados bajo la acepción de *structure from motion*, capaces de extraer información tridimensional de un objeto si se dispone de varias tomas del mismo a lo largo del tiempo, siempre que se impongan restricciones apropiadas.

En visión estéreo, la información que necesitamos para reconstruir información tridi-

mensional de la escena son “correspondencias” entre las vistas de la escena, es decir, poder saber qué puntos de una imagen de la escena corresponden a qué puntos en la otra imagen. Asimismo, en el caso de las técnicas de *structure from motion*, la información necesaria es la misma, y justamente es la información de que disponemos en nuestro caso, ya que precisamente seguimos *features* de los vehículos a lo largo de su historia en la secuencia, de modo que podemos establecer correspondencias en los puntos que ocupan dichas características.

El disponer de un post-proceso que reconstruyera la forma tridimensional de los vehículos permitiría poder catalogarlos (utilitario, familiar, monovolumen, deportivo, etc), proveyendo de una fuente de información sobre la que hacer minería de datos.

5.2.5. *Object recognition*

Otra aproximación para la identificación de vehículos es la denominada *object recognition* ([19, 18]). Este tipo de técnicas intentan identificar los vehículos a partir de sus partes: las *features* discretas extraídas de los vehículos de la escena se comparan con las de modelos previamente almacenados para averiguar el tipo de vehículo de que se trata. Este tipo de aproximaciones permiten implementar técnicas de *eigen-tracking*, utilizando una aproximación de optimización para encontrar la “solución” dentro del espacio de estados.

La implementación de este tipo de mecanismo como añadido a nuestro algoritmo, permitiría de esta manera reconocer el tipo de vehículos, lo que abriría las puertas a un nuevo tipo de funcionalidad: el “conocer” a los vehículos que pasan por ciertos puntos kilométricos, unido a una hipotética infraestructura distribuida, permitiría “trazar el recorrido” que van haciendo los coches, así como obtener una visión más holística de la velocidad a la que circulan, y no sólo en el campo de acción local a cada radar, como ocurre en la actualidad, ya que se podría conocer la velocidad media entre puntos de control de velocidad consecutivos.

5.2.6. Infraestructura de monitorización de tráfico

Por último, un paso de mayor nivel en la expansión del método desarrollado, sería aprovechar su facilidad de despliegue para establecer una infraestructura global de monitorización de tráfico: actualmente la Dirección General de Tráfico, ha creado una *capa* de *Google maps* que solapa información acerca de las carreteras sobre los mapas nacionales (véase figura 5.1). El despliegue masivo de elementos de control de la velocidad del tráfico permitiría añadir a dicha capa información detallada de la densidad del tráfico, velocidad media, etc, en cada tramo de trazado.

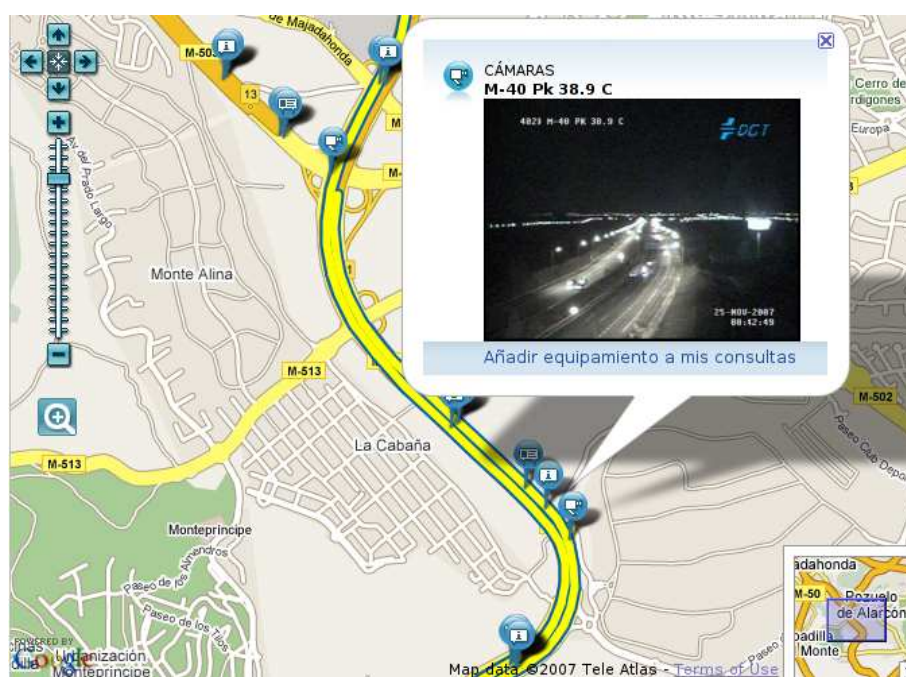


Figura 5.1: Información de tráfico solapada a *Google maps* en la web de la DGT.

Bibliografía

- [1] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt. Performance of optical flow techniques. *CVPR*, 92:236–242.
- [2] D. Beymer and J. Malik. Tracking vehicles in congested traffic. In A. C. Chachich and M. J. de Vries, editors, *Proc. SPIE Vol. 2902, p. 8-18, Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*, Alan C. Chachich; Marten J. De Vries; Eds., volume 2902 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 8–18, February 1997.
- [3] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters, 1997.
- [4] A. Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag, 1998.
- [5] O. Faugeras. *Three-dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.
- [6] Robert Grover Brown and Patrick Y.C. Hwang. *Introduction to random signals and applied Kalman filtering*. John Wiley and Sons, Inc., 3rd edition, 1997.
- [7] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of 4th Alvey Vision Conf., Manchester*, pages 147–151, 1988.
- [8] B.K.P. Horn and B.G. Schunck. Determining optical flow. 17(1-3):185–203, August 1981.
- [9] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [11] D. Koller, K. Daniilidis, and H.H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *IJCV*, 10(3):257–281, June 1993.

- [12] X.-R. Li and V. P. Jilkov. Survey of maneuvering target tracking: dynamic models. In O. E. Drummond, editor, *Proc. SPIE Vol. 4048, p. 212-235, Signal and Data Processing of Small Targets 2000, Oliver E. Drummond; Ed.*, pages 212–235, July 2000.
- [13] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.
- [14] Bruce D. Lucas. *Generalized Image Matching by the Method of Differences*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 1984.
- [15] Perez Lucena, Fuertes. Using optical flow for tracking.
- [16] Enrique Muñoz Corral. Calibración de cámara con corrección de la distorsión radial. Trabajo fin de carrera, Facultad de Informática UPM, diciembre 2000.
- [17] M.Zuliani, C.Kenney, and B.S.Manjunath. A mathematical comparison of point detectors. In *Proceedings of CVPRW*, 2004.
- [18] E.Ñowak and F. Jurie. Vehicle categorization: Parts for speed and accuracy. pages 277–283, 2005.
- [19] V.S. Petrovic and T.F. Cootes. Analysis of features for rigid structure vehicle type recognition. pages xx–yy, 2004.
- [20] P. F. Sturm and S. J. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *Proceedings of CVPR*, pages 432–437, 1999.
- [21] An-Ting Tsao, Yi-Ping Hung, Chiou-Shann Fuh, and Yong-Sheng Chen. Ego-motion estimation using optical flow fields observed from multiple cameras.
- [22] M.B. van Leeuwen and F.C.A. Groen. Egomotion estimation for traffic applications.
- [23] Greg Welch and Gary Bishop. *Introduction to the Kalman filter*. Dept.of Computer Science, University of North Carolina at Chapel Hill, April 2004.
- [24] Zhengyou Zhang. A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Corporation, One Microsoft Way.Redmond, WA 98052, December 1998.
- [25] A. Zisserman and R. Hartley. *Multiple View Geometry*. Cambridge, Oxford, England, 1st edition, 2000.